Computer Simulations of High Energy Physics

Philip John Stephens

Trinity College

A dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy
May 2004

Computer Simulations of High Energy Physics

Philip John Stephens

Abstract

This thesis describes the development of two independent computer programs, Herwig++ and Effective. Both of these programs are used for phenomenological predictions of high energy physics. The former is used to simulate events as measured at particle colliders. The latter is used to generate the mass spectrum of supersymmetric models.

Simulation of collider events requires the implementation of several different aspects of particle phenomenology. After a brief introduction on the relevant aspects of the Standard Model and numerical techniques, a new set of variables for parton shower evolution are presented. These new variables retain the angular ordering feature of the variables found in the original HERWIG software, while improving the Lorentz invariance of the shower and improving the coverage of the phase space. These new variables define new initial conditions for the shower depending on how the partons are colour connected. Also developed is a new model for hadronization. By changing the distribution of probabilities of cluster decays into hadron pairs this model is able to enforce desired results, such as isospin symmetry or meson-baryon ratios, more intuitively.

The physics of the Herwig++ software is described in detail. The improvements to the new evolution variables and the new hadronization model provide are illustrated by comparing them against data for e^+e^- events.

Effective is a program that is able to provide the 1-loop effective potential and 1-loop mass matrices for an arbitrary $N = 1$ supersymmetric model. This software also is able to solve the renormalization group equations at one-loop for the parameters of the model and, in turn, provide the scale dependent values of these parameters. The program is described and some results indicative of its potential are also presented.

Declaration

This dissertation is the result of my own work, except where explicit reference is made to the work of others, and has not been submitted for another qualification to this or any other university.

Philip John Stephens

Preface

This thesis contains the work I have done on Herwig++ and Effective during the course of my high-energy physics PhD. The first chapter is an original summary of theoretical and practical issues pertaining to my research. The results of chapter 2 were derived independently by Prof. Bryan Webber and myself and cross checked for errors. Chapter 3 is the result of my own independent work. Chapter 4 is a complete discussion of Herwig++ which has been implemented by myself and Dr. Stefan Gieseke, with other coding contributions from Dr. Alberto Ribon. Theoretical contributions to Herwig++ have also been made by Prof. Bryan Webber and Dr. Mike Seymour. Chapter 5 is the result of comparison studies of Herwig++ to LEP data. This is the result of work done in conjunction by Dr. Gieseke and myself. Chapter 6 is the extension of work done by Dr. James Hetherington as part of his PhD thesis. His original idea of Effective has been extended to include the renormalization group equations and the correct one-loop mass corrections have been implemented.

I would like to thank Dr. Stefan Gieseke for his continual help and guidance throughout my time at Cambridge. I would also like to thank my supervisor Prof. Bryan Webber and the rest of the members of the HEP Group over the last three years, in particular Dr. Agustin Sabio Vera and Dr. Jeppe Anderson for numerous fruitful discussions.

The work of chapter 6 would not have been possible if not for the time and energy spent by Dr. James Hetherington helping me to understand the ideas he had implemented in Effective.

I would like to thank Trinity College and the University Rugby Club for membership

and support. Also the Universities UK and the Cambridge Overseas trust for their financial support. And I'd like to thank CERN for allowing me to spend a summer there while attending the Monte Carlo Workshop for LHC, during which discussions with the SHERPA team helped me to understand the content of chapter 3 better.

Mostly I would like to thank my family, old and new. Thanks to my parents for encouragement, support and belief that I would achieve this goal. Thanks to my brothers and sister for constantly supporting me and for their confidence in my abilities. Lastly I would like to thank Jenna. This is the first step in our long life of happiness together.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Field Theory Introduction

This section briefly introduces a few key ideas that are used throughout high energy physics for calculating predictions of physics. I start by introducing the Klein-Gordon and Dirac field equations. This is followed by a discussion of the *Lagrangian* for both *Abelian* and *non-Abelian* gauge groups. Lastly I explain how calculations are performed in *perturbation theory*.

We start by introducing the Klein-Gordon field. This field obeys Bose statistics and is used to describe all bosons: scalars and vectors. Free fields of this form obey the Klein-Gordon equation

$$(\partial^2 + m^2)\phi(x) = 0. \tag{1.1}$$

The propagator of a field is the Green's function of the field equation. In this case it is the Green's function of the Klein-Gordon equation. In order to find this function we must introduce a pole prescription for the integral. Figure 1.1 shows the contour used for the integral over p^0. When the contour is closed below we find the Green's function

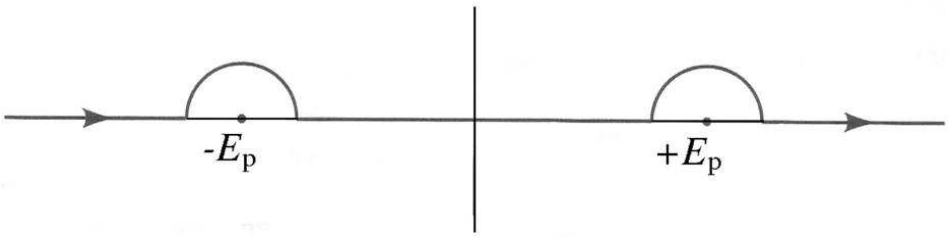


Figure 1.1: The contour taken to find the retarded Green's function. For $x_0 > y_0$ we can close the contour below. For $x_0 < y_0$ we can close the contour above, giving zero.

only over the range $x_0 > y_0$. This is called the *retarded* Green's function. For $x^0 > y^0$ it is

$$D_R(x) = \int \frac{d^4p}{(2\pi)^4} \frac{ie^{-ip\cdot(x-y)}}{p^2 - m^2}. \tag{1.2}$$

and zero for $x_0 < y^0$.

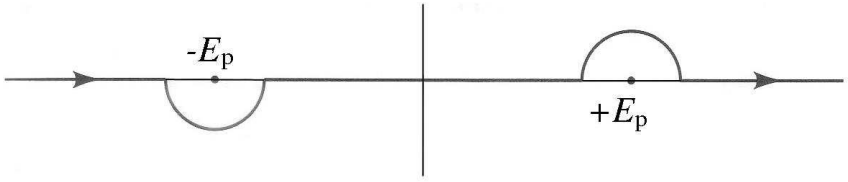


Figure 1.2: The contour for the Feynman prescription. When $x_0 > y_0$ the contour can be closed below and when $x_0 < y_0$ the contour can be closed above.

Using the *Feynman prescription* of pole contours we find the propagator, known as the *Feynman propagator*, for bosonic fields over all x^0 and y^0 is

$$D_F(x-y) \equiv \int \frac{d^4p}{(2\pi^4)} \frac{e^{-ip\cdot(x-y)}}{p^2 - m^2 + i\epsilon}. \tag{1.3}$$

This can be written as

$$D_F(x-y) = \begin{cases} \langle 0|\, \phi(x)\phi(y)\, |0\rangle & \text{for } \; x^0 > y^0 \\ \langle 0|\, \phi(y)\phi(x)\, |0\rangle & \text{for } \; x^0 < y^0 \end{cases}. \tag{1.4}$$

We now want to see how fields that obey Fermi statistics behave. The Dirac Equation is the field equation for spin 1/2 fermions. We first start by introducing the Dirac matrices γ^μ. These are in four-dimensional Minkowski space and are given in a popular representation by

$$\gamma^0 = \begin{pmatrix} 0 & \mathbf{1} \\ \mathbf{1} & 0 \end{pmatrix}; \quad \gamma^i = \begin{pmatrix} 0 & \sigma^i \\ -\sigma^i & 0 \end{pmatrix}, \tag{1.5}$$

where $\mathbf{1}$ is the 2×2 identity matrix and σ^i are the Pauli sigma matrices. Using these matrices we define the following notation: $\not{a} \equiv \gamma^\mu a_\mu$ and $\bar{\psi} \equiv \psi^\dagger \gamma^0$. The Dirac equation is then

$$(i\not{\partial} - m)\psi(x) = 0. \tag{1.6}$$

Doing the same as for the Klein-Gordon equation, we can find that the retarded Green's function, in momentum space, of a Dirac field is

$$\tilde{S}_R = \frac{i}{\not{p} - m} = \frac{i(\not{p} + m)}{p^2 - m^2}. \tag{1.7}$$

Along with the anti-commuting nature of fermions this leads to the Feynman propagator

$$S_F(x - y) = \int \frac{d^4p}{(2\pi)^4} \frac{i(\not{p} + m)}{p^2 - m^2 + i\epsilon} e^{-ip\cdot(x-y)}, \tag{1.8}$$

which is

$$S_F(x - y) = \begin{cases} \langle 0| \psi(x)\bar{\psi}(y) |0\rangle & \text{for } x^0 > y^0 \\ -\langle 0| \bar{\psi}(y)\psi(x) |0\rangle & \text{for } x^0 < y^0 \end{cases}. \tag{1.9}$$

1.1.1 Lagrangian

In classical mechanics, the fundamental quantity is the action, S. This is the time integral of the Lagrangian L. The quantity L can be written as a spatial integration of a Lagrangian density, $\mathcal{L}$. It is $\mathcal{L}$ that is generally referred to as the Lagrangian in field

theory. This is also done so throughout this thesis.

The Lagrangian contains the information about all of the fields and interactions of a theory. The different types of interactions are mediated by particles which transform according to the adjoint representation of a gauge group. In the Standard Model (SM), these particles are gauge bosons; spin 1 particles. The kinetic term for a gauge boson, $\mathcal{A}^A_\mu$, is

$$\mathcal{L}_{\text{gauge}} = -\frac{1}{4} F^A_{\mu\nu} F^{A\mu\nu}, \tag{1.10}$$

where

$$F^A_{\mu\nu} = \partial_\mu \mathcal{A}^A_\nu - \partial_\nu \mathcal{A}^A_\mu - g f^{ABC} \mathcal{A}^B_\mu \mathcal{A}^C_\nu. \tag{1.11}$$

In equation (1.11) the indices A, B and C are the indices in the adjoint representation of the group and the indices μ and ν are Lorentz indices. g is known as a *coupling constant*. These define the relative scale of the term in the Lagrangian when compared to other terms. The terms f^{ABC} are the structure constants of the gauge group. Each group has a set of generators t^A. The structure constants for the gauge group are given by the relation

$$\left[t^A, t^B\right] = i t^C f^{ABC}. \tag{1.12}$$

When f^{ABC} is equal to 0 for all indices the group is called an Abelian group. When they are non-zero it is called a non-Abelian group. From (1.11) it can be seen that for an Abelian group the gauge fields don't interact with themselves. For quantum electrodynamics (QED) the group is $U(1)$ which is an Abelian group. The gauge field of QED is the photon and therefore we can see that photons don't interact with themselves. For quantum chromodynamics (QCD) this group is $SU(3)$, which is non-Abelian. It is the non-Abelian nature of this group that gives rise to the triplet and quartic gluon self-interaction terms which complicate QCD calculations.

In order for the symmetries of a gauge group to be preserved, all the terms in the

Lagrangian must be invariant under the local gauge transformations of that group. The Lagrangian is composed of gauge fields, matter fields and derivatives of both. As the Lagrangian needs to be invariant under these transformations, only certain combinations of these fields are possible.

The matter fields furnish the fundamental representation of the group. In the Standard Model these are fermions or scalars; spin $\frac{1}{2}$ or spin 0 particles. For a set of local transformations, $\theta^A(x)$, we have the transformation

$$\Psi(x)_i \to \left(\mathrm{e}^{i\sum_A t^A\theta^A(x)}\right)_{ij}\Psi(x)_j \equiv \Omega(x)_{ij}\Psi(x)_j. \tag{1.13}$$

Here, the indices i and j indicate the fundamental indices and the index A is the index of the adjoint representation of the group. In $SU(N)$ the adjoint representation runs from 1 to (N^2-1) and N is the number of fundamental indices. In $U(N)$ the adjoint indices run from 1 to N^2. For example in $SU(2)$ there are 3 adjoint index values and 2 fundamental index values. Likewise in $SU(3)$ there are 8 adjoint index values and 3 fundamental index values.

We now define the *covariant derivative* for a local transformation. This is

$$D_\mu \equiv \partial_\mu + ig\sum_A t^A\mathcal{A}^A_\mu(x). \tag{1.14}$$

Here g is the coupling constant of the group. As mentioned before, these govern the relative strength the interactions of one group have when compared with the other groups. In order to use the Klein-Gordon and Dirac field equations in the Lagrangian we must replace the partial derivate ∂_μ by the covariant derivative D_μ. We also require the covariant derivative to transform in the same way as the matter fields. This is the requirement

$$D_\mu\Psi(x)_i \to \Omega(x)_{ij}D_\mu\Psi(x)_j. \tag{1.15}$$

We can use these transformations to define the transformation of the gauge fields. In the Abelian case this is

$$\mathcal{A}_\mu(x) \to \mathcal{A}_\mu(x) - \frac{1}{g}\partial_\mu\theta(x), \tag{1.16}$$

while the non-Abelian case the transformation is more complicated. It is given by

$$\sum_A t^A \mathcal{A}^A_\mu \to \Omega(x)\left(\sum_A t^A \mathcal{A}^A_\mu\right)\Omega^{-1}(x) + \frac{i}{g}\left(\partial_\mu\Omega(x)\right)\Omega^{-1}(x). \tag{1.17}$$

This requirement prevents the addition of a term like $\frac{1}{2}m^2\mathcal{A}_\mu\mathcal{A}^\mu$ to give the field mass. Instead this must be done by the Higgs mechanism, which is described in section 1.2.2.

The coupling constants entered into the Lagrangian are known as *bare couplings*. Due to higher order corrections, these are not the same as the physically observed couplings. Instead the physical coupling is a *renormalized coupling*. This means that the higher order corrections introduce a shift in the coupling.

At first it may seem that there are an infinite number of possible terms in the Lagrangian that can satisfy gauge invariance. It will be shown in Section 1.1.2 that higher order terms in perturbation theory will involve integrals over 4-momenta of virtual particles. These are divergent integrals. In order to do the calculations these must have some cutoff imposed at finite momentum, Λ. Theories where the observables, expressed in terms of the suitably renormalized parameters, have values which are independent on Λ are known as *renormalizable* theories. Using this it can be shown that theories which contain a coupling constant of mass to the *negative* power are not renormalizable.

The last constraint on the Lagrangian is that it must have dimension $(\text{mass})^4$. Using the constraints of gauge invariance, renormalizability and the correct dimension of the Lagrangian the only allowable terms in the Lagrangian, for spinors ψ, scalars ϕ and

vectors $\mathcal{A}_\mu$, are

$$\bar{\psi}\psi, \quad \phi^\dagger\phi, \quad (\phi^\dagger\phi)^2, \quad \bar{\psi}\psi\phi,$$
$$\bar{\psi}\gamma^\mu D_\mu\psi, \quad (D_\mu\phi)^\dagger(D^\mu\phi), \quad F_{\mu\nu}F^{\mu\nu}.$$

All of these terms can have some relevant coupling that has mass dimension larger than or equal to 0. For a more thorough discussion of gauge invariance, renormalizability and Lagrangian formalism see [1–6].

1.1.2 Perturbation Theory

Earlier we found the Feynman propagator for a field that obeys the Klein-Gordon equation. This can be written as

$$D_F(x-y) = \int \frac{d^4p}{(2\pi)^4} \frac{i\mathrm{e}^{-ip\cdot(x-y)}}{p^2 - m^2 + i\epsilon} \equiv \langle 0 \left| T\left\{\phi(x)\phi(y)\right\}\right| 0\rangle , \tag{1.18}$$

where T indicates the time-ordering property of the Feynman propagator. This propagator describes the free field theory. Physical predictions can only be made on theories that interact, however. To do so we work in the *interaction picture*. We now define a unitary operator, $U(t, t_0)$, that takes a field at time t_0 to time t in the presence of an interaction. This is known as the interaction picture propagator and is defined as

$$U(t, t_0) = \mathrm{e}^{iH_0(t-t_0)}\mathrm{e}^{-iH(t-t_0)}, \tag{1.19}$$

where we have divided the Hamiltonian into the free field part, H_0 and the interaction part H_{int}. We find U obeys the Schrödinger equation

$$i\frac{\partial}{\partial t}U(t, t_0) = \mathrm{e}^{iH_0(t-t_0)}(H - H_0)\mathrm{e}^{-iH(t-t_0)} = H_I(t)U(t, t_0). \tag{1.20}$$

H_I is the interaction Hamiltonian written in the interaction picture. This has the form

$$H_I(t) = \mathrm{e}^{iH_0(t-t_0)} \left(H_{\mathrm{int}}\right) \mathrm{e}^{-iH_0(t-t_0)}. \tag{1.21}$$

Solving this differential equation with the initial condition $U(t_0, t_0) = 1$ we can find the solution as

$$U(t, t_0) = T\left\{\exp\left[-i\int_{t_0}^{t} dt' H_I(t')\right]\right\}, \tag{1.22}$$

where the time-ordering of an exponential is the Taylor series with each term time ordered. It is this Taylor series that is used when doing perturbative calculations. Before we can define these perturbative calculations we must introduce *Wick's Theorem*.

In the interaction picture we can decompose the field ϕ into its positive and negative energy parts

$$\phi(x) = \phi^+(x) + \phi^-(x). \tag{1.23}$$

The *contraction* of two fields is defined as

$$\underline{\phi(x)\phi(y)} \equiv \begin{cases} [\phi^+(x), \phi^-(y)] & \text{for } x^0 > y^0; \\ [\phi^+(y), \phi^-(x)] & \text{for } y^0 > x^0. \end{cases} \tag{1.24}$$

This is exactly the Feynman propagator that was encountered before

$$\underline{\phi(x)\phi(y)} = D_F(x - y). \tag{1.25}$$

This allows us to write the time-ordering of fields as

$$T\left\{\phi(x_1)\phi(x_2)\ldots\phi(x_m)\right\} = N\left\{\phi(x_1)\phi(x_2)\ldots\phi(x_m) + \text{all possible contractions}\right\}, \tag{1.26}$$

where N indicates the *normal ordering*. This is just the ordering of having all creation

operators to the right of all annihilation operators. The identity in (1.26) is Wick's theorem.

When computing the vacuum expectation value of a time-ordered product any uncontracted operators from applying Wick's theorem give zero ($\langle 0 | N(\text{any operator}) | 0 \rangle = 0$) and the contracted operators are simply Feynman propagators! It is this decomposition of the time ordered products that leads to Feynman diagrams.

When we wish to construct higher order terms we will have states that are created and destroyed and never produce observable particles. These are known as *virtual* particles. When computing observables, the contributions due to these particles must be integrated over their momenta. As we can see from (1.3) and (1.8), these integrals contain p^{-2} for each virtual particle and d^4p for the integrals over these particles. Using power counting it can be shown that some diagrams will contain integrals like

$$\int^{\Lambda} \frac{dp}{p} \sim \ln(\Lambda) . \tag{1.27}$$

These integrals need to be bound above[a] by an ultra-violet cutoff, Λ, in order to be computed. Observables must be independent of this cutoff and this condition is what leads to renormalizability, as discussed previously.

The power of perturbation theory is fully exploited when the coupling constants are small. This allows us to write (1.22) as a Taylor series in order of the coupling constant. Using Wick's theorem, we can then decompose the terms of the Taylor series into normal ordered products. This allows us to use Feynman diagrams to describe each of the normal ordered products. Where perturbation theory is a good approximation we only need to evaluate the first few terms of the series in order to approximately describe the physics. Unfortunately, the magnitude of the higher order terms cannot be

[a]These also need to be bound below, known as an infrared cutoff. This is another problem that is not related to renormalizability.

predicted beforehand. Only by calculating them can one decide how accurate the initial calculation really is.

1.2 The Standard Model

The Standard Model (SM) is well established as a model that describes the particles and all the interactions, except gravity. The predictions of the model have been tested to high accuracy by the series of LEP experiments. Recent experimental evidence [7] shows that neutrino flavours oscillate which means they must have mass. This is in direct contradiction to the SM and is the first evidence of physics beyond the SM. Apart from the incorrect description of the neutrino flavour oscillation, the model also predicts the existence of the Higgs boson. This has not been experimentally confirmed to date. This section explains the particle content of the model, the Higgs mechanism and properties of QCD.

There are theoretical reasons to believe that at higher energies the Standard Model will also break down. In section 6.1 I discuss what shortcomings there are believed to be and a theoretical solution to these shortcomings, known as supersymmetry (SUSY).

1.2.1 Particle Content

The Standard Model is composed of the $SU(3)_c, SU(2)_L$ and the $U(1)_Y$ gauge groups. Properties of the interaction eigenstates are governed by these groups. The weak and and electromagnetic interactions are not directly governed by $SU(2)_L$ and $U(1)_Y$ groups, however. Instead the mass eigenstates are given by the symmetry breaking of these two groups. This will be discussed in section 1.2.2.

The gauge fields for these groups are $\mathcal{A}^a_\mu$, W^i_μ and B_μ. The field $\mathcal{A}$ is known as

the gluon field. As mentioned earlier, these gluons have self-interaction terms. That means a gluon, unlike the neutral photon, carries a (colour) charge. These terms make calculations with QCD much more complex than with QED. The W and B bosons mix through the Higgs mechanism (see section 1.2.2) to form the $W^{\pm}$, Z^0 bosons and the photon, A[b]. It is the photon that mediates the electromagnetic interaction we observe. For each field we have a term in the Lagrangian given by (1.11).

The matter content of the Standard Model can be summarized quite simply. There are leptons, quarks and the Higgs boson. Leptons have $SU(2)_L$ and $U(1)_Y$ charges and the quarks have $SU(3)_c$, $SU(2)_L$ and $U(1)_Y$ charges. The L subscript of the $SU(2)$ gauge means that it only couples to left handed particles. We can see from table 1.1 that there are right and left handed charged leptons, up-type quarks and down-type quarks but there are no right handed neutrinos. This means that the left handed leptons interact with the W and B bosons, but not with the gluons, $\mathcal{A}$, while the right handed leptons only interact with the B boson. The left and right handed quarks interact with the gluons and the B boson. The left handed quarks also interact with the W bosons while the right handed quarks don't. Table 1.1 shows the charges of the fields in each gauge. As will be discussed later, the Higgs boson has $SU(2)_L$ and $U(1)_Y$ charges.

The strength of the different interactions (QED, Weak, QCD) are dictated by the relative size of their coupling constants. In QED this constant is e which is related to the $SU(2)_L$ coupling, g_W and the $U(1)_Y$ coupling, g'. In weak interactions the $SU(2)_L$ symmetry is broken and the coupling depends on whether we are coupling to the $W^{\pm}$ bosons or the Z^0 boson. Either way this coupling is smaller then e, thus the weak interaction is weaker than the electromagnetic one. The coupling constant for QCD is g_3. This is larger than e; for fields that interact with gluons the QCD terms are most often the dominant ones.

[b]Note the different font between the gluon field and the photon field.

Field	$\mathrm{U}(1)_Y$	$\mathrm{SU}(2)_L$	$\mathrm{SU}(3)_c$
e_R	-1	**1**	**1**
ℓ	$-\frac{1}{2}$	**2**	**1**
Q	$\frac{1}{6}$	**2**	**3**
u_R	$\frac{2}{3}$	**1**	**3**
d_R	$-\frac{1}{3}$	**1**	**3**

Table 1.1: The fermionic fields of the Standard Model and their charges. There are left handed $\mathrm{SU}(2)_L$ doublets, ℓ and Q, and right handed $SU(2)_L$ singlets, e_R, u_R and d_R. There is also three families of each type of fermion.

There is more to this picture than the couplings being constant, however. In fact these terms are scale dependent. This is known as the running of the couplings and will be discussed in section 1.2.3. It is believed that at some large scale, the couplings of all the interactions, including gravity, will be of the same size, thus unifying the theories.

It is known from experiments that these particles have mass. In the SM the fermions have right and left handed components which have different charges under the groups. Due to gauge invariance, a term like $m\bar{\psi}_L\psi_R$ cannot be added to the Lagrangian to give these fields mass as they have different $SU(2)_L$ and $U(1)_Y$ charges. Instead, the Higgs mechanism is used again to define a Yukawa coupling. This will also be explained in the next section.

1.2.2 Higgs Mechanism

The Higgs mechanism is a mechanism for generating the masses of the Standard Model particles while keeping the Lagrangian gauge invariant. The idea hinges on a scalar field being added to the model which has a non-zero vacuum expectation value (VEV). The scalar can then couple to the particles in the Standard Model and by doing so defines their masses. Because the SM Higgs boson has charges under $SU(2)_L$ and $U(1)_Y$ the non-zero VEV breaks the gauge invariance. This is why the B boson of the $U(1)_Y$ group

is not the mediator of the electromagnetic interaction. Instead the photon is, which is composed of both the B boson and the W_3 boson.

The SM Higgs boson field, Φ, is a $\mathrm{SU}(2)_L$ doublet and carries $\mathrm{U}(1)_Y$ charge. This means that it couples to the W bosons and the B boson through the covariant derivative. The field can be defined as

$$\Phi = U(\xi) \begin{pmatrix} 0 \\ \frac{1}{\sqrt{2}}(v + H) \end{pmatrix}, \tag{1.28}$$

where $U(\xi)$ is a unitary operator in $SU(2)$ with three degrees of freedom. These three degrees of freedom are known as the Goldstone bosons. It is these bosons that are 'eaten' in order to give the $W^{\pm}$ and Z^0 bosons mass. Though these terms do not appear as physical particles they do play a role in calculations depending on the choice of gauge fixing term, ξ. H is a real scalar field which has a zero VEV and is interpreted as the physical Higgs boson field.

The potential of the Higgs boson is given by a linear combination of $\Phi^\dagger\Phi$ term and a $(\Phi^\dagger\Phi)^2$ term. The result is added to the Lagrangian in the SM. The Lagrangian for the Higgs boson is

$$\mathcal{L}_{\text{Higgs}} = (D^\mu\Phi)^\dagger D_\mu\Phi + \mu^2\Phi^\dagger\Phi - \lambda\left(\Phi^\dagger\Phi\right)^2. \tag{1.29}$$

The positive sign in front of the μ^2 is different from the standard mass term in a Lagrangian. Therefore, when the parameters $\mu^2, \lambda > 0$ this gives the potential a minimum at $\langle\Phi\rangle \neq 0$. The μ and λ parameters also dictate what the VEV of the field is that minimizes the potential. The particular combination

$$\langle\Phi\rangle \equiv \frac{v}{\sqrt{2}} = \frac{\mu}{\sqrt{2\lambda}}, \tag{1.30}$$

defines the value of the VEV that minimizes the potential. Figure 1.3 shows the potential

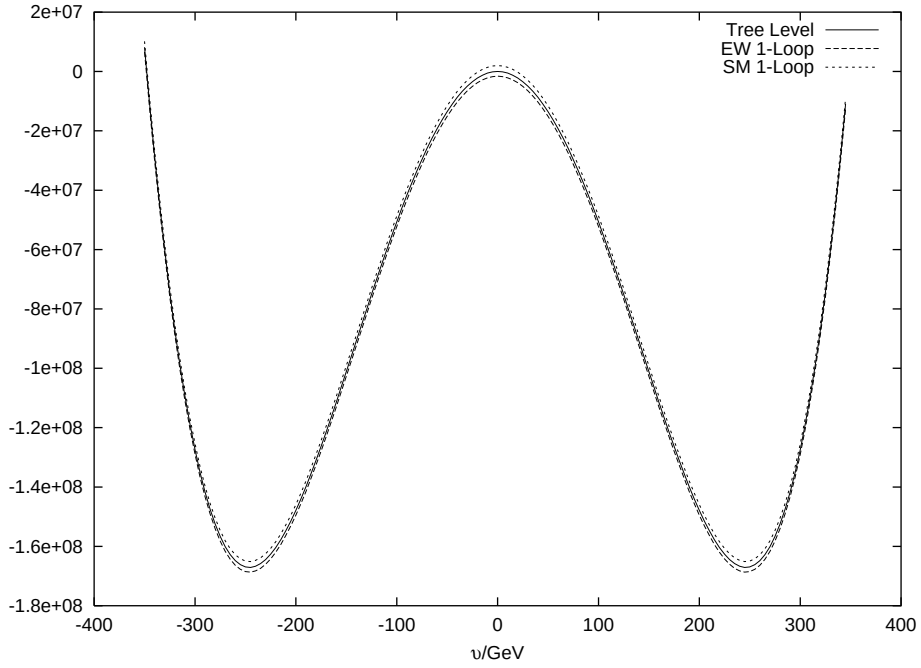


Figure 1.3: The effective potential of the Standard Model as υ is varied. The three lines are the tree level potential, the 1-Loop correction only with the Electroweak particles (leptons, W,Z,γ,Higgs boson) and the one-loop correction for all Standard Model particles. These are all for the combination $\frac{\mu}{\sqrt{\lambda}} = 246.0$ GeV

as the VEV is varied for a choice of $\frac{\mu}{\sqrt{\lambda}} = 246.0$ GeV. It can be seen that at $\upsilon = 246.0$ GeV this potential is a minimum. Also in the figure is the 1-loop effective potential including only the electroweak (EW) particles and the 1-loop effective potential when all the SM particles are included. These are also minimized at the same value of the VEV. This figure was generated by putting the model into the software Effective, which will be explained in Chapter 6.

The covariant derivative term, $\mathcal{L}_{\text{Cov Higgs}} = (D_\mu\Phi)^*(D^\mu\Phi)$ from (1.29), when expanded out is

$$\mathcal{L}_{\text{Cov Higgs}} = \left| \begin{pmatrix} \partial_\mu - \frac{i}{2}(g_W W_\mu^3 + g' B_\mu) & -\frac{i}{2}\left(W_\mu^1 - iW_\mu^2\right) \\ -\frac{i}{2}\left(W_\mu^1 + iW_\mu^2\right) & \partial_\mu - \frac{i}{2}(-g_W W_\mu^3 + g' B_\mu) \end{pmatrix} \begin{pmatrix} 0 \\ \frac{\upsilon+H}{\sqrt{2}} \end{pmatrix} \right|^2 . \tag{1.31}$$

From this term we find that the W^3 boson and the B boson mix. This is known as

spontaneous symmetry breaking as the exact symmetries $\mathrm{SU}(2)_L$ and $\mathrm{U}(1)_Y$ are broken down to $U(1)_{\mathrm{em}}$ and a broken $SU(2)_W$. We can see the $SU(2)_W$ is broken as the $W^{\pm}$ bosons and the Z^0 have different masses. The mixing between W_3 and B produces mass eigenstates with mass squared $\frac{1}{4}\left(g'^2 + g_W^2\right)v^2$ and 0. These are then interpreted as the Z^0 boson and the photon, A, respectively. This means that the mass eigenstates are defined as

$$\begin{pmatrix} Z^{0\mu} \\ A^{\mu} \end{pmatrix} = \begin{pmatrix} \cos\theta_W & -\sin\theta_W \\ \sin\theta_W & \cos\theta_W \end{pmatrix} \begin{pmatrix} W_3^{\mu} \\ B^{\mu} \end{pmatrix}. \tag{1.32}$$

where $\cos^2\theta_W = \frac{g_W^2}{g_W^2+g'^2}$ and $\sin^2\theta_W = \frac{g'^2}{g_W^2+g'^2}$. Here θ_W is known as the Weinberg angle. The mass squared of the $W^{\pm}$ boson is $\frac{1}{4}g_W^2v^2$, where the W^+ bosons is the combination

$$W^{\mu+} = \frac{1}{\sqrt{2}}(W_1^{\mu} - iW_2^{\mu}), \tag{1.33}$$

and $W^{\mu-}$ is the complex conjugate.

The electric charge, Q, of a field is then given by $eQ = eT_3 + eY$. T_3 is the eigenvalue of the third generator of the $SU(2)_L$ group. For a $\mathrm{SU}(2)_L$ doublet this is $\frac{1}{2}$ for the first component of the doublet and $-\frac{1}{2}$ for the second component. e is given as $e = g_W \sin\theta_W = g'\cos\theta_W$. We can see from table 1.1 that the charged leptons have charge -1, neutrinos have charge 0, up-type quarks have charge $+\frac{2}{3}$ and down-type quarks have charge $-\frac{1}{3}$.

The value of the VEV can be determined from the Fermi constant, G_F. This is measured as 1.166×10^{-5} GeV^{-2}. It is defined as

$$\frac{g_W^2}{8M_W^2} \equiv \frac{1}{2v^2} = \frac{G_F}{\sqrt{2}} \tag{1.34}$$

and yields the value $v = 246$ GeV. This value, along with the value of θ_W then correctly predicts the mass of the Z^0 boson as well. The fact that this mechanism predicts the

masses of the Z^0 and $W^{\pm}$ bosons as well as the fact that the photon is massless, in a gauge invariant way, has led to the belief that the Higgs boson must exist. Fortunately, even though the particle hasn't been found yet its mass is given by $m_H = \sqrt{2}\mu$. Since μ is only restricted by loop corrections it gives bounds of the Higgs mass [8] between 117 GeV and 251 GeV with 95% confidence; experimentally this ceiling has not yet been reached. Finding the Higgs boson is one of the main goals of the upcoming Large Hadron Collider (LHC).

The Higgs boson also allows a gauge invariant way to introduce fermion masses as well. Instead of having a term $m_x\bar{\psi}_{xL}\psi_{xR}$ for each fermionic field x we introduce instead a Yukawa matrix Y^f_{ij} for each fermionic family (leptons and quarks). The Yukawa term in the Lagrangian is

$$\mathcal{L}_{\text{Yukawa}} = Y^f_{ij}\bar{\psi}_{fL}\Phi\psi_{fR} + \text{h.c.}, \tag{1.35}$$

where the subscript L and R denote the left and right handed fields. The VEV of Φ then generates the mass terms for all the fermions and the mixing between families. Of course, we still have independent parameters to define the mass of the fields.

The most general gauge invariant term that can be added for the quark masses is

$$\mathcal{L}_m = -Y^d_{ij}\bar{Q}_{ia}\phi_a d_{Rj} - Y^u_{ij}\epsilon^{ab}\bar{Q}_{ia}\phi^\dagger_b u_{Rj} + \text{h.c.}. \tag{1.36}$$

The Yukawa matrices are not necessarily symmetric or Hermitian. In fact, there is no principle that even requires that they are real valued! However, if CP is conserved, this would be true. We can simplify the form of (1.36) by diagonalizing the matrices obtained from squaring the Yukawa matrices, Y_u, Y_d. Each one defines two unitary matrices, U_i and W_i, for $i = u, d$ by

$$Y_i Y_i^\dagger = U_i D_i^2 U_i^\dagger, \quad Y_i^\dagger Y_i = W_i D_i^2 W_i^\dagger, \tag{1.37}$$

where D_i is the diagonal matrix with elements which are the positive square roots of the eigenvalues. The Yukawa matrices can then be defined as

$$Y_i = U_i D_i W_i^\dagger. \tag{1.38}$$

If we now make a chiral rotation of the right handed fields by W and the left handed fields by U, these transformations don't affect the couplings of these particles to the Higgs field. In this basis, we also find that P, C and T are conserved.

Since all the up and down type quarks have the identical couplings in QCD the U matrices commute with the covariant derivative of QCD. This transformation, however, mixes u_L and d_L and does affect the $SU(2) \times U(1)$ couplings. If we now neglect the QCD interactions and write the Lagrangian in the basis of the Z^0, $W^\pm$ and A, rather than the W's and B, we have

$$\begin{aligned}
\mathcal{L} &= \bar{\ell}(i\not{\partial})\ell + \bar{e}_R(i\not{\partial})e_R + \bar{Q}(i\not{\partial})Q + \bar{u}_R(i\not{\partial})u_R + d_R(i\not{\partial})d_R \\
&\quad + g\left(W_\mu^+ J_W^{\mu+} + W_\mu^- J_W^{\mu-} + Z_\mu^0 J_Z^\mu\right) + eA_\mu J_{\rm EM}^\mu,
\end{aligned} \tag{1.39}$$

where

$$\begin{aligned}
J_W^{\mu+} &= \frac{1}{\sqrt{2}}\left(\bar{\nu}_L\gamma^\mu e_L + \bar{u}_L\gamma^\mu d_L\right); \\
J_W^{\mu-} &= \frac{1}{\sqrt{2}}\left(\bar{e}_L\gamma^\mu \nu_L + \bar{d}_L\gamma^\mu u_L\right); \\
J_Z^\mu &= \frac{1}{\cos\theta_W}\left[\bar{\nu}_L\frac{\gamma^\mu}{2}\nu_L + \bar{e}_L\gamma^\mu\left(-\frac{1}{2} + \sin^2\theta_W\right)e_L + \bar{e}_R\gamma^\mu\sin^2\theta_W e_R\right. \\
&\quad + \bar{u}_L\gamma^\mu\left(\frac{1}{2} - \frac{2}{3}\sin^2\theta_W\right)u_L + \bar{u}_R\gamma^\mu\left(-\frac{2}{3}\sin^2\theta_W\right)u_R \\
&\quad \left. + \bar{d}_L\gamma^\mu\left(-\frac{1}{2} + \frac{1}{3}\sin^2\theta_W\right)d_L + \bar{d}_R\gamma^\mu\left(\frac{1}{2}\sin^2\theta_W\right)d_R\right]; \\
J_{\rm EM}^\mu &= -\bar{e}\gamma^\mu e + \frac{2}{3}\bar{u}\gamma^\mu u - \frac{1}{3}\bar{d}\gamma^\mu d.
\end{aligned} \tag{1.40}$$

These currents contain an abundance of information. For example, the first two show that the $W^{\pm}$ couple up-type quarks to down-type quarks and neutrinos to charged leptons. From the electromagnetic current one can directly read off the electric charges of the different particles.

We can now see how these currents change when the chiral transformations, U and W, are applied. We can see that in the electromagnetic current, J^{μ}_{EM}, the transformation matrices cancel out

$$\bar{d}^i_L \gamma^\mu d^i_L \to \bar{d}^i_L \left(U^\dagger_d\right)^{ij} \gamma^\mu U^{jk}_d d^k_L = \bar{d}^i_L \gamma^\mu d^i_L. \tag{1.41}$$

This is also true for J^μ_Z. The current that couples to the $W^{\pm}$, however, does change. This is

$$J^{\mu+}_W = \frac{1}{\sqrt{2}} \bar{u}^i_L \gamma^\mu d^i_L \to \frac{1}{\sqrt{2}} \bar{u}^i_L \gamma^\mu \left(U^\dagger_u U_d\right)^{ij} d^j_L. \tag{1.42}$$

This defines a new matrix

$$V = U^\dagger_u U_d, \tag{1.43}$$

which is known as the *Cabibbo-Kobayashi-Maskawa* (CKM) mixing matrix. This explains why strange quarks enter into weak interactions. The $W^{\pm}$ boson is able to not only turn up-type quarks into down-type quarks but also change the generation in the process.

The same arguments that were just given can also be applied to the lepton families. Since the neutrinos don't interact in any way except by the weak interactions we can by convention choose to label the mass eigenstates of the neutrinos according to the charged lepton partner it is formed with. Unlike the case of the quarks, there is no way to distinguish these states in another way. This way a $W^{\pm}$ boson only couples the neutrinos of one generation to a charged lepton of the same generation.

1.2.3 Running Coupling

We start by defining α for a theory as $\alpha = \frac{g^2}{4\pi}$. In QED g is the electric charge of the positron and the α is the fine structure constant and is denoted simply by α. In QCD g is g_3 and the α is labelled α_S. In order to remove the ultra-violet divergences in the perturbative series a renormalization procedure is used. This procedure introduces a mass scale μ. Therefore, when we want to calculate a dimensionless physical observable at mass scale Q, it can only depend on the ratio Q^2/μ^2, which is not constant. The choice of μ is arbitrary and therefore if we were to hold the bare coupling of the Lagrangian fixed a physical quantity, R, cannot depend on μ. Instead it must depend only on Q^2/μ^2 and the renormalized couplings. We will consider first the running of α_S, as it plays an important role in QCD. In QCD this can all be expressed as

$$\mu^2 \frac{d}{d\mu^2} R(Q^2/\mu^2, \alpha_S) \equiv \left[\mu^2 \frac{\partial}{\partial \mu^2} + \mu^2 \frac{\partial \alpha_S}{\partial \mu^2} \frac{\partial}{\partial \alpha_S} \right] R = 0. \tag{1.44}$$

Identifying

$$t = \ln\left(\frac{Q^2}{\mu^2}\right), \quad \beta(\alpha_S) = \mu^2 \frac{\partial \alpha_S}{\partial \mu^2}, \tag{1.45}$$

the μ independence of the observable R for massless particles can be expressed as

$$\left[-\frac{\partial}{\partial t} + \beta(\alpha_S) \frac{\partial}{\partial \alpha_S} \right] R\left(\mathrm{e}^t, \alpha_S\right) = 0. \tag{1.46}$$

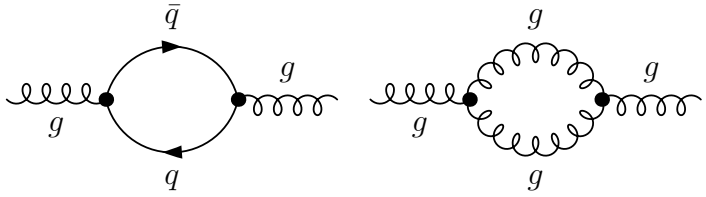
It is from this that we define the *running coupling* $\alpha_S(Q^2)$ as

$$t = \int_{\alpha_S}^{\alpha_S(Q^2)} \frac{dx}{\beta(x)}, \quad \alpha_S(\mu^2) \equiv \alpha_S. \tag{1.47}$$

We can then see that

$$\frac{\partial \alpha_S(Q^2)}{\partial t} = \beta(\alpha_S(Q^2)), \quad \frac{\partial \alpha_S(Q^2)}{\partial \alpha_S} = \frac{\partial \beta(\alpha_S(Q^2))}{\partial \beta(\alpha_S)}. \tag{1.48}$$

A solution to (1.46) is $R(1, \alpha_S(Q^2))$. Therefore, the scale dependence of R is due solely to the running of the coupling. By the same means we can find the running coupling of α for QED interactions.



(a) (b)

Figure 1.4: Graphs which contribute to the QCD β function in the one-loop approximation (in a physical gauge).

The running coupling constants are then determined by the renormalization group equation (RGE),

$$Q^2 \frac{\partial \alpha}{\partial Q^2} = \beta(\alpha). \tag{1.49}$$

These always have at least α^2 dependence as they are extracted from higher-order loop corrections to the bare vertices of a theory. Fig. 1.4 shows contributions to the β function for QCD at one-loop. Figure 1.4b shows a new interaction due to the non-Abelian nature of QCD. The β function in QCD then has the perturbative expansion

$$\beta(\alpha) = -b\alpha_S^2 \left(1 + b'\alpha_S + b''\alpha_S^2 + O(\alpha_S^3)\right). \tag{1.50}$$

The coefficient b'' depends on the renormalization scheme. b and b' don't, however, and are given by

$$\begin{aligned} b &= \frac{11C_A - 2n_f}{12\pi} \\ b' &= \frac{17C_A^2 - 5C_A n_f - 3C_F n_f}{2\pi(11C_A - 2n_f)}, \end{aligned} \tag{1.51}$$

where $C_A = 3$, $C_F = \frac{4}{3}$ and n_f is the number of active flavours. From this we see that in order for b to be negative (which corresponds to the leading term of the expansion being positive) we must have $n_f \geq \frac{11C_A}{2}$. For $C_A = 3$ this is $n_f \geq 16$. If we look at the QED β function

$$\beta_{\text{QED}}(\alpha) = \frac{1}{3\pi}\alpha^2 + O(\alpha^3). \tag{1.52}$$

The leading term of this β is always positive. This is where we can see the difference the non-Abelian interactions (triplet and quartic gluon vertices) of QCD makes. QCD is known as an *asymptotically free* theory. This means that α_S becomes smaller as Q^2 increases, corresponding to the opposite sign of the β function.

A fuller discussion of RGEs is given in chapter 6 as a part of the development of the software Effective.

1.2.4 Asymptotic Freedom

We start first by explaining why the observed charge of the electron decreases with distance. Referring back to our previous discussion of the running coupling we see that as Q^2 increases the QED coupling increases. This corresponds to it growing at small distances. This can be easily explained as follows. The larger the distance between the observer and the charge means that more electron-positron pairs will be temporarily created out of the vacuum. These can be considered temporary electric dipoles which are preferentially aligned with the positive end towards the electron and the negative end away. Fig. 1.5 shows this situation. This effectively screens the bare charge of the electron and it appears to have a smaller charge the further away from the electron you are.

In QCD, the picture is not quite so clear. We now have three degrees of colour charge, as opposed to one in QED, and non-Abelian interactions. There are two ways

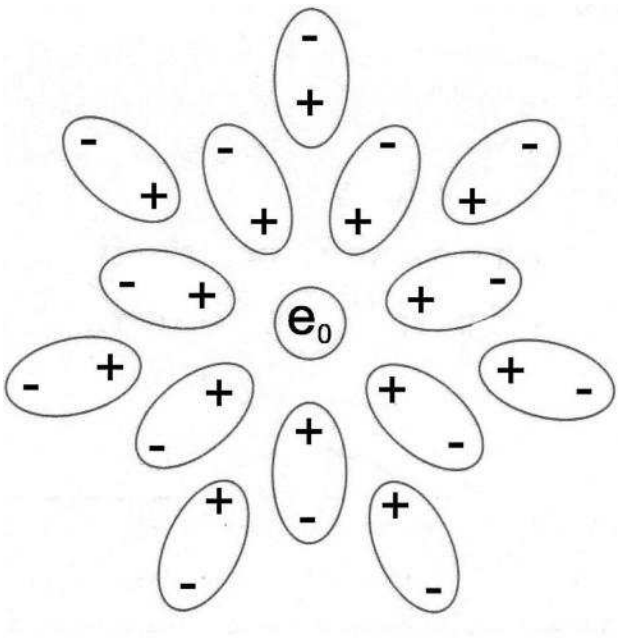


Figure 1.5: Virtual e^+e^- pairs are effectively dipoles that screen the bare charge of the electron.

of describing the difference: either as a dielectric effect or as a paramagnetic effect.

We start by giving the dielectric effect argument. We can define a running charge at scale q^2 in terms of the charge at an ultra-violet cutoff $\Lambda_{\rm UV} \gg q^2$ and a scale-dependent dielectric constant, $\epsilon(q^2)$. This is

$$\alpha(q^2) = \frac{\alpha\left(\Lambda^2_{\rm UV}\right)}{\epsilon(q^2)} \tag{1.53}$$

and from the previous discussion we find

$$\frac{1}{\epsilon(q^2)} = 1 - \frac{\beta(\alpha)}{\alpha} \ln \frac{\Lambda^2_{\rm UV}}{q^2}. \tag{1.54}$$

This implies that the running charge satisfies the equation

$$\frac{d\alpha(q^2)}{d\ln q^2} = \beta(\alpha). \tag{1.55}$$

We see that in QED β is positive meaning the dielectric constant is greater than one. This corresponds to a screening of the charge. In QCD we have the opposite case. Here the dielectric constant is smaller than one. This is an antiscreening of the charge.

If we now assume the vacuum of a relativistic quantum field theory can be treated as a polarizable medium we can define a magnetic permeability, $\mu(q^2)$ that due to Lorentz invariance must satisfy the equation

$$\epsilon(q^2)\mu(q^2) = 1 \tag{1.56}$$

for all q. We can now analyze the behavour of the magnetic susceptibility, $\chi \equiv \mu - 1$. From (1.54) we have

$$\chi(q^2) = -\frac{\beta(\alpha)}{\alpha} \ln \frac{\Lambda_{\text{UV}}^2}{q^2}. \tag{1.57}$$

This can be broken up into two parts. The first term, known as Pauli paramagnetism, describes how the spins interact with the magnetic field and the second term, known as the Landau diamagnetism, describes how the orbital motion of the particles interacts with the magnetic field. When the Pauli paramagnetic term is larger then the Landau diamagnetic term the system is considered a paramagnetic system, otherwise it is diamagnetic. In QCD the Pauli paramagnetism term has some dependence on both spin $\frac{1}{2}$ quarks and spin 1 gluons. The higher spin gluons make a larger contribution than the quarks. The permeability is given by (1.57) when

$$\beta(\alpha) = -b\alpha^2. \tag{1.58}$$

In QCD a contribution from a particle with spin S to b is given by

$$b = \frac{(-1)^{2S}}{2\pi} \left[(2S)^2 - \frac{1}{3} \right]. \tag{1.59}$$

This gives a total b term in QCD of

$$b = \frac{11C_A - 2n_f}{12\pi}. \tag{1.60}$$

From this we can see that the permeability can only be negative (making it a diamagnetic system) when $n_f > 16$. This shows that QCD is asymptotically free due to the colour charge carrying spin 1 gluons.

QCD is an asymptotically free theory. It also has a coupling which runs. The running decreases α_S as Q^2 increases and therefore high energy QCD allows the methods of perturbation theory to be applied.

This is by no means an exhaustive reference of the Standard Model and more complete descriptions are readily available [1–3, 9, 10]. This section has, however, described the basics of the particle content, the Higgs mechanism and two topics of importance in QCD: running couplings and asymptotic freedom. The existence of the Higgs boson is taken as an assumption for the development of SUSY, and will be done so for the remainder of this thesis.

1.3 Event Generation

We now move into the discussion of Monte Carlo Event Generators. Chapters 2 through 5 are all discussions on the development of the event generator Herwig++. In this section I start by introducing the Monte Carlo integration technique and explain why this is so useful for high-energy particle physics simulations. I then introduce all the various parts of an event: hard process, parton shower, hadronization and hadron decays.

1.3.1 Monte Carlo Approach

The idea of the Monte Carlo approach is that the value of an integral can be calculated using random numbers. The same ideas are also applicable to sampling based on a distribution. Many calculations of quantum field theory involve matrix elements where the amplitude squared is interpreted as a probability; the Monte Carlo approach is an excellent fit for computer simulations due to these probability distributions. This is because points can be drawn according to a distribution, thus simulating a physical event with the correct probabilities. A more thorough discussion of Monte Carlo integration is given in [11].

1.3.1.1 Simple Monte Carlo Integration

If we pick N random points which are uniformly distributed in a multidimensional volume V, the basic theorem of Monte Carlo integration is that the integral of a function f over the multidimensional volume can be approximated by

$$\int f dV \sim V \left(\langle f \rangle \pm \sqrt{\frac{\langle f^2 \rangle - \langle f \rangle^2}{N}} \right). \tag{1.61}$$

The angle brackets indicate the arithmetic mean over the N points in V space,

$$\langle f \rangle \equiv \frac{1}{N} \sum_{i=1}^{N} f(x_i) \qquad \langle f^2 \rangle \equiv \frac{1}{N^2} \sum_{i=1}^{N} f^2(x_i). \tag{1.62}$$

The "plus-minus" term in (1.61) indicates an estimation of one standard deviation of the integral.

To calculate the value of the integral using (1.61), one simply generates a set of x values in the volume V which are the argument of $f(x)$. Using the two parts of (1.62) and the set of values generated the approximate value of the integral is given.

There is also an algorithm known as the rejection algorithm which is useful when generating values according to a distribution. In this case we don't want to calculate the integral but rather generate a point according to the distribution. This can be done by simply generating the points uniformly in a volume which is V plus an extra dimension bounded by the function. If the point generated lies inside the function we accept the point. Otherwise we reject it and generate a new one. If we wanted to take the integral, this would then be the percentage of accepted points times the volume sampled over. If we consider a function of one variable, $f(x)$, we would generate N pairs of points, (x, y). We find the ratio of points where $y < f(x)$, and multiply this ratio by the area the points are generated in. Figure 1.6 shows an example of this technique. This technique is called integration by rejection. In the example above the points are generated in an unweighted manner. It is also possible to make the algorithm more efficient or add some more information by generating points in a weighted manner as well.

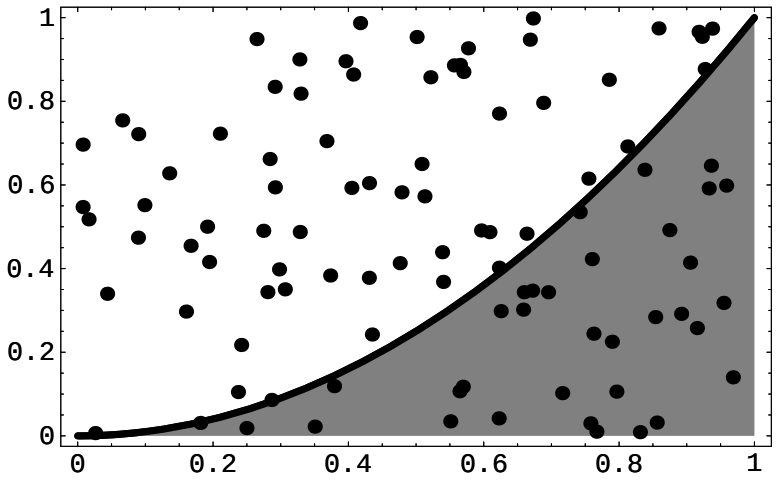


Figure 1.6: This shows the randomly distributed points in a square of area 1. The value of the integral is the percentage of points under the function

The method can easily be expanded to integrate over regions with an unknown volume. Say we want to integrate over a strange region, C, of which we do not know the volume. This can also be easily done with the Monte Carlo technique. Simply expand the region the points are generated in to a simpler region, M, which contains C, and

the multi-dimensional volume of M is known. When the point lies outside of the desired region, set the value of the function to 0.

1.3.1.2 Non-Uniform Sampling for Monte Carlo Integration

When working with probability distributions, we will be using the points that lie inside of the function to be a physical quantity. Therefore, in the simple approach, there may be many points that are generated, that aren't used in the simulation. In particle physics, many of this distributions have very sharp peaks and valleys. These peaks and valleys cause a large number of points to be generated which aren't used. This is a large waste of CPU time and we would like to optimize this.

Fortunately, this can be done. If we instead generate our points according to some other distribution, rather than a uniform one, we can reduce the number of unused points. Optimally, we would want to generate them according to the function being integrated, but this would require that we know the value of the integral beforehand! Instead, we find another function that we can integrate, that is always larger than the function we would like to integrate. We want to choose this function so that the sharp peaks and valleys of our unknown integral are also, approximately, present in our known function. But, in order to sample according to our new function, it must be invertible.

The evaluation of the integral with this new distribution is not much different than for a uniform distribution. Again, we will consider a function of one variable, $f(x)$. Our known function is $g(x)$ with integral G. Now we randomly generate the x's and evaluate both $f(x)$ and $g(x)$. A random number, $\mathcal{R}$ between 0 and 1 is also generated. If $\mathcal{R}$ is is less than the ratio $f(x)/g(x)$, then the point is under the function. Otherwise it is not. The value of the integral is then just G times the percentage of the points under the function. This technique can greatly reduce the number of unused points that are generated. The improvement is dependent on how well the known function, g, estimates

$f(x)$. An example of this is shown in Figure 1.7.

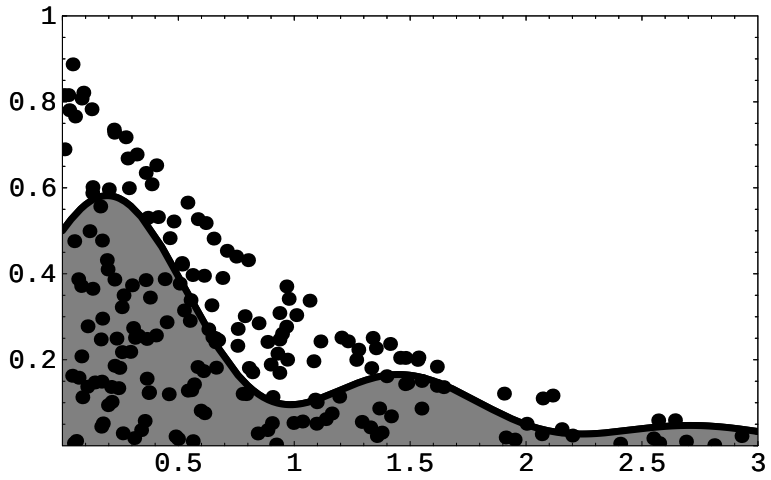


Figure 1.7: This shows the random numbers sampled under a known function, $g(x)$. The area of the integral is now G, the integral under $g(x)$, times the percent of points under $f(x)$.

Luckily, there are some general features for QCD matrix elements that can be used to take advantage of non-uniform sampling. These are well known [12, 13] and general algorithms have been developed to improve the efficiency of generating points for these matrix elements.

There are also some other optimizations that can be performed, such as stratified sampling. One program that is particularly useful for particle physics integrals is VEGAS [14]. More detail on these optimizations is also given in [11].

1.3.2 Hard Process

There are two separate momentum regimes that different methods of high energy physics can be used at. The first is the perturbative regime. This regime is for high momentum transfer, short distances. In this regime calculations can be approximated by truncating the time-ordered series at any order and applying Wick's theorem. Calculations are usually calculated in orders of α (couplings). The *hard process* fits into this regime.

There is also the non-perturbative regime. In this regime calculations are complicated to perform because as we saw in Section 1.1.2 calculations depend on a time-ordered exponential and each term is itself a complicated expression. As was shown earlier, the value of α_S is largest in the low momentum transfer regime. This means the calculations would need to be performed to many orders of α_S. Currently calculations to next-to-next-to leading order (leading order + 2 more orders) are state-of-the-art. In fact, in the low momentum transfer regime one of the tools of perturbative physics, the free field propagator is not valid anymore. Since non-perturbative calculations are so difficult, models of physics are usually used instead of trying exact calculations. The process of *hadronization* is an example of this.

The hard process is the underlying process that occurs when two beams of particles are collided. The description of what happens is divided into two different parts, the actual hard process and the Parton Distribution Functions (PDFs). A similar division in this discussion is used. Here I discuss the hard process and in the next section explain the PDFs and how they affect the hard process.

The hard process of an event is what describes the interaction of high energy particles that 'collide'. There are numerous processes for a given set of incoming particles. For example, in an e^+e^- collision we could produce another e^+e^- pair through a $e^+e^- \rightarrow \gamma^* \rightarrow e^+e^-$ or a $e^+e^- \rightarrow Z^0 \rightarrow e^+e^-$ process. We could also produce a $q\overline{q}$ pair in similar fashion. In fact, there is an infinite number of possible final states as any number of photons or gluons can be emitted by an electrically or colour charged particle. The higher order terms fall off rapidly however and contribute only a fraction of the total cross section. The actual process that occurs is proportional to the fraction of the total cross section for a given process.

In event generators a specific set of hard processes can be used, without considering all possibilities. This way different properties particular to certain hard process can be

studied. Given a set of processes, one is chosen based on the ratio of each cross section to the total cross sections in the allowable set. At the lowest order of perturbation theory, there are relatively few processes to calculate for a given pair of incoming particles. As the next order of perturbation will contain a new factor of the coupling, these processes are suppressed. It is impossible to know beforehand what the cross section for the next order will be; it must be calculated. In particular regions of phase space (soft and collinear) there is a formalism for the splitting of a particle, a, into particles $b+c$. These can then be used to generate approximations to higher order diagrams. More detail on this is given in Section 1.3.4. These splitting functions are then used to generate the QED and QCD emissions from a parton. This process is known as the *parton shower*.

The development of matrix elements for a given process is a complete discussion in itself [3,9,10]. Here we only give the results of the main processes studied in this thesis. The first is $e^+e^- \to f\overline{f}$ [4]. The differential cross section for this process with either a γ or Z^0 in the s-channel and θ the center-of-mass scattering angle of the outgoing fermions is

$$\begin{aligned}\frac{d\sigma}{d\cos\theta} &= \frac{\pi\alpha^2}{2s}\left[(1+\cos\theta)\left\{Q_f^2 - 2Q_f V_e V_f F^{\gamma}_{Z^0}(s)\right.\right. \\ &\quad \left. +(A_e^2+V_e^2)(A_f^2+V_f^2)F^{Z^0}_{Z^0}(s)\right\} \\ &\quad \left. +\cos\theta\left\{-4Q_f A_e A_f F^{\gamma}_{Z^0}(s) + 8A_e V_e A_f V_f F^{Z^0}_{Z^0}(s)\right\}\right], \qquad (1.63)\end{aligned}$$

where

$$\begin{aligned}F^{\gamma}_{Z^0}(s) &= \frac{\sqrt{2}G_F M_Z^2}{16\pi\alpha}\frac{s(s-M_Z^2)}{(s-M_Z^2)^2+\Gamma_Z^2 M_Z^2}, \\ F^{Z^0}_{Z^0}(s) &= \frac{2G_F^2 M_Z^4}{256\pi^2\alpha^2}\frac{s^2}{(s-M_Z^2)^2+\Gamma_Z^2 M_Z^2}. \qquad (1.64)\end{aligned}$$

s is the centre-of-mass energy and A_f and V_f are the vector and axial couplings of

fermion f to the Z^0 boson. These are

$$V_f = T_f^3 - 2Q_f \sin^2\theta_W, \quad A_f = T_f^3. \tag{1.65}$$

The functions $F_{Z^0}^{\gamma}(s)$ and $F_{Z^0}^{Z^0}(s)$ are the contributions from the Z^0-γ interference and the Z^0-exchange, respectively. This process is of particular importance when comparing the simulations to data taken at LEP. For center-of-mass energies well below the Z^0 peak, the two functions $F(s)$ can be ignored yielding a differential cross section of

$$\frac{d\sigma}{d\cos\theta} = \frac{\pi\alpha^2 Q_f^2}{2s}(1+\cos^2\theta). \tag{1.66}$$

Integrating over θ gives a total cross section of

$$\sigma_0 = \frac{4\pi\alpha^2 Q_f^2}{3s}. \tag{1.67}$$

Around the Z pole, the $F_{Z^0}^{Z^0}(s)$ term dominates and the cross section is approximately

$$\sigma_0 = \frac{G_F M_Z^2}{96\pi\Gamma_Z^2}(A_e^2+V_e^2)(A_f^2+V_f^2). \tag{1.68}$$

To use simulations for hadron-hadron events another set of matrix elements is used. Below are the matrix elements squared that have been spin and colour averaged (summed) over the initial (final) states for massless partons [4].

$$\begin{aligned}
\tfrac{1}{g_3^4}\bar{\sum}|\mathcal{M}|^2_{qq'\to qq'} &= \frac{4}{9}\frac{\hat{s}^2+\hat{u}^2}{\hat{t}^2};\\
\tfrac{1}{g_3^4}\bar{\sum}|\mathcal{M}|^2_{q\bar{q}'\to q\bar{q}'} &= \frac{4}{9}\frac{\hat{s}^2+\hat{u}^2}{\hat{t}^2};\\
\tfrac{1}{g_3^4}\bar{\sum}|\mathcal{M}|^2_{qq\to qq} &= \frac{4}{9}\left(\frac{\hat{s}^2+\hat{u}^2}{\hat{t}^2}+\frac{\hat{s}^2+\hat{t}^2}{\hat{u}^2}\right)-\frac{8}{27}\frac{\hat{s}^2}{\hat{u}\hat{t}};\\
\tfrac{1}{g_3^4}\bar{\sum}|\mathcal{M}|^2_{q\bar{q}\to q'\bar{q}'} &= \frac{4}{9}\frac{\hat{t}^2+\hat{u}^2}{\hat{s}^2};
\end{aligned}$$

$$
\begin{aligned}
\frac{1}{g_3^4}\bar{\sum}|\mathcal{M}|^2_{q\bar{q}\to q\bar{q}} &= \frac{4}{9}\left(\frac{\hat{s}^2+\hat{u}^2}{\hat{t}^2}+\frac{\hat{t}^2+\hat{u}^2}{\hat{s}^2}\right)-\frac{8}{27}\frac{\hat{u}^2}{\hat{s}\hat{t}};\\
\frac{1}{g_3^4}\bar{\sum}|\mathcal{M}|^2_{q\bar{q}\to gg} &= \frac{32}{27}\frac{\hat{t}^2+\hat{u}^2}{\hat{t}\hat{u}}-\frac{8}{3}\frac{\hat{t}^2+\hat{u}^2}{\hat{s}^2};\\
\frac{1}{g_3^4}\bar{\sum}|\mathcal{M}|^2_{gg\to q\bar{q}} &= \frac{1}{6}\frac{\hat{t}^2+\hat{u}^2}{\hat{t}\hat{u}}-\frac{3}{8}\frac{\hat{t}^2+\hat{u}^2}{\hat{s}^2};\\
\frac{1}{g_3^4}\bar{\sum}|\mathcal{M}|^2_{gq\to gq} &= -\frac{4}{9}\frac{\hat{s}^2+\hat{u}^2}{\hat{s}\hat{u}}+\frac{\hat{u}^2+\hat{s}^2}{\hat{t}^2};\\
\frac{1}{g_3^4}\bar{\sum}|\mathcal{M}|^2_{gg\to gg} &= \frac{9}{2}\left(3-\frac{\hat{t}\hat{u}}{\hat{s}^2}-\frac{\hat{s}\hat{u}}{\hat{t}^2}-\frac{\hat{s}\hat{t}}{\hat{u}^2}\right),
\end{aligned}
\tag{1.69}
$$

where $\bar{\sum}$ is the spin averaged sum and $\hat{s},\hat{t},\hat{u}$ are the usual Mandelstam variables for the process $AB\to CD$: $\hat{s}\equiv(p_A+p_B)^2$, $\hat{t}\equiv(p_A-p_C)^2$ and $\hat{u}\equiv(p_A-p_D)^2$.

1.3.3 Parton Distribution Functions

Hadrons are composed of quarks. This means that when two hadrons are collided, it is some components of the hadrons that interact fundamentally. Determining which part of the hadrons are the ones that interact is a complex issue. This is defined by the *Parton Distribution Functions*. These are developed from the data of deep inelastic scattering (DIS) experiments. This section provides a brief discussion of these functions. For a more detailed discussion, see [4, 15, 16]. We also discuss how the remaining parts of the hadrons are handled in a Monte Carlo event generator. These are called the *beam remnants*.

The *factorization theorem* allows the study of the parton constituents to be factorized into a non-perturbative part and a perturbative part. The non-perturbative part is determined from experiments, while the perturbative part can be calculated as a perturbation series ordered in the strong coupling constant, α_S.

We start by considering the DIS process $ep\to eX$. Figure 1.8 shows this process with the relavant momenta k, q and p. We start by introducing the variables

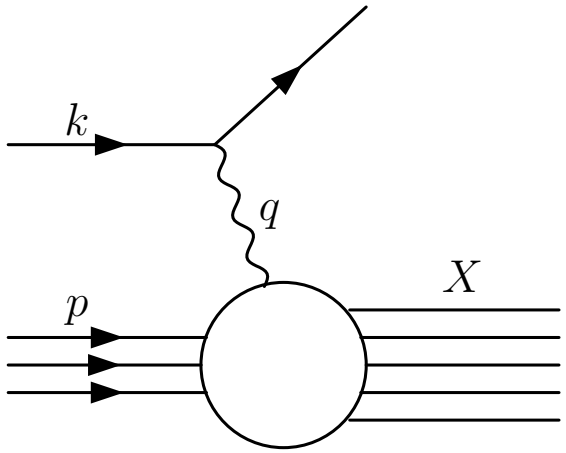


Figure 1.8: The electron emits a virtual photon which probes the structure of the proton.

$$Q^2 = -q^2, \quad \nu = p \cdot q, \quad x = \frac{Q^2}{2\nu}, \quad y = \frac{p \cdot q}{p \cdot k}. \tag{1.70}$$

We now define the hadronic tensor, $W^{\mu\nu}$. This defines how the hadron will interact with the photon. This is given by

$$W^{\mu\nu}(p,q) = \frac{1}{4} \sum_r \sum_X \langle p, r \left| J^\mu(0) \right| X \rangle \langle X \left| J^\nu(0) \right| p, r \rangle (2\pi)^3 \delta^4(p + q - P_X), \tag{1.71}$$

where J^μ is the electromagnetic current and r is the spin of the proton. The variable X is summed over all of the possible final products of the process. If we require that this tensor conserves parity and we have unpolarized protons, we find that we can decompose this unknown tensor into two independent amplitudes, $F_1(x, Q^2)$ and $F_2(x, Q^2)$, called structure functions. We now write this tensor as

$$W^{\mu\nu}(p,q) = -\left(g^{\mu\nu} + \frac{q^\mu q^\nu}{Q^2}\right) F_1(x, Q^2) + \frac{1}{\nu}\left(p^\mu + \frac{\nu}{Q^2} q^\mu\right)\left(p^\nu + \frac{\nu}{Q^2} q^\nu\right) F_2(x, Q^2). \tag{1.72}$$

The total differential cross section for this process in terms of these structure functions is

$$\frac{d^2\sigma}{dxdy} = \frac{4\pi\alpha^2}{xyQ^2}\left((1-y)F_2 + xy^2F_1 - \frac{m^2}{Q^2}x^2y^2F_2\right). \tag{1.73}$$

The *Bjorken limit* is defined as $Q^2, \nu \to \infty$ with x fixed. In this limit the structure functions depend, approximately, only on x. This implies the photons scatter off *pointlike constituents*. If we work in the 'infinite momentum frame' we can ignore the mass of the proton. Comparing (1.73) to the spin averaged matrix element for the process $e^-q \to e^-q$ yields the relation

$$\hat{F}_2 = xe_q^2\delta(x-\xi) = 2x\hat{F}_1, \tag{1.74}$$

where the hat indicates that the stucture function refers to a quark, not a proton and the ξ in this context is the fraction of the protons momentum that the quark constituent carries. Measurements show that the momentum is not given by a delta function but by a distribution. This means that the quarks carry a range of momentum fractions.

The ideas generated by studying the structure functions in the Bjorken limit are incorporated into the 'naive parton model'. In this model the following assumptions are made:

- $f_i(\xi)d\xi$ is the probability that quark of flavour i carries a momentum fraction between ξ and $\xi + d\xi$,
- the photon scatters incoherently off the quark constituents.

(1.74) can then be written as

$$\begin{aligned} F_2(x) = 2xF_1(x) &= \sum_i \int_0^1 d\xi f_i(\xi) x e_q^2 \delta(x-\xi) \\ &= \sum_i e_q^2 f_i(x). \end{aligned} \tag{1.75}$$

Beyond leading order the 'naive' parton model is broken in QCD by logarithms of Q^2. In the Bjorken limit the transverse momentum is assumed to be small. The higher order contributions show that is is not the case though. A quark is able to emit a gluon with large transverse momentum with probability $\alpha_S \frac{dk_T^2}{k_T^2}$ at large k_T. These contributions give terms proportional to $\alpha_S \log(Q^2)$. It is these terms that break scaling. When Q^2 is large enough, these terms compensate for the small value of α_S. The approximation where these terms are summed to all orders is known as the leading-log approximation (LLA). It is this approximation that the parton shower is developed at.

The breaking of the scaling means that the functions $f_i(x)$ gain a scale dependence and are instead $f_i(x, Q^2)$. These can be written at some renormalization scale μ as

$$\begin{aligned} f_i(x,\mu^2) &= f_{i0}(x) + \frac{\alpha_S}{2\pi}\int_x^1 \frac{d\xi}{\xi}\left[f_{i0}(\xi)\left\{P_{qq}\left(\frac{x}{\xi}\right)\ln\frac{\mu^2}{\kappa^2} + C_q\left(\frac{x}{\xi}\right)\right\}\right. \\ &\quad \left. + f_{g0}(\xi)\left\{P_{qg}\left(\frac{x}{\xi}\right)\ln\frac{\mu^2}{\kappa^2} + C_g\left(\frac{x}{\xi}\right)\right\}\right] + \ldots, \end{aligned} \tag{1.76}$$

where κ is an infrared cutoff and f_g is a new function, similar to f_i but describing a gluon rather than a quark. P_{qq} is a splitting function in the case of a quark emitting a gluon before scattering off the virtual photon, whereas the P_{qg} is a splitting function for the case of a gluon splitting into a $q\bar{q}$ pair and the q scattering off the virtual photon. $f_{i0}(x)$ is an unmeasurable, bare distribution. This distribution absorbs all of the collinear singularities at a 'factorization scale' μ. $F_2(x, Q^2)$ can then be written in terms of (1.76) by

$$\begin{aligned} F_2(x,Q^2) &= x\sum_{i=q,\bar{q}} e_q^2 \int_x^1 \frac{d\xi}{\xi}\left[f_i(\xi,\mu^2)\left\{\delta(1-\frac{x}{\xi}) + \frac{\alpha_S}{2\pi}P_{qq}\left(\frac{x}{\xi}\right)\ln\frac{Q^2}{\mu^2} + \ldots\right\}\right. \\ &\quad \left. + f_g(\xi,\mu^2)\left\{\frac{\alpha_S}{2\pi}P_{qg}\left(\frac{x}{\xi}\right)\ln\frac{Q^2}{\mu^2} + \ldots\right\}\right]. \end{aligned} \tag{1.77}$$

This has absorbed all of the finite contribution, C, into the parton distributions. It is

possible to factor out an arbitrary finite term from the distributions which leaves behind an additional finite contribution. This depends on the 'factorization scheme'. A common choice is the $\overline{\text{MS}}$ scheme in which (1.77) is expressed as

$$\begin{aligned} F_2(x,Q^2) &= x\sum_{i=q,\bar{q}} e_q^2 \int_x^1 \frac{d\xi}{\xi}\left[f_i(\xi,Q^2)\left\{\delta(1-\frac{x}{\xi}) + \frac{\alpha_S}{2\pi}C_q^{\overline{\text{MS}}}\left(\frac{x}{\xi}\right)+\dots\right\}\right. \\ &\quad \left. + f_g(\xi,Q^2)\left\{\frac{\alpha_S}{2\pi}C_g^{\overline{\text{MS}}}\left(\frac{x}{\xi}\right)+\dots\right\}\right]. \end{aligned} \tag{1.78}$$

The functions $C_q(x)$ and $C_g(x)$ are called the *coefficient functions* and they depend on the factorization and renormalization schemes. These are the perturbative part of the PDFs. The functions $f_i(x,\mu^2)$ contain all the non-perturbative physics.

Though the parton distributions are not derivable in the scope of perturbation theory, we can see from (1.77) that the right side cannot be dependent on μ^2. Taking the partial derivative of both sides with respect to $t=\mu^2$, and ignoring the gluon part for simplicity, yields the DGLAP equation [17–20]

$$t\frac{\partial}{\partial t}f_i(x,t) = \frac{\alpha_S(t)}{2\pi}\int_x^1 \frac{d\xi}{\xi}P\left(\frac{x}{\xi}\right) f_i(\xi,t). \tag{1.79}$$

This equation is analogous to the β functions which describe the variation of $\alpha_S(t)$ with t. Including all the terms this can more generally be written as a $(2n_q+1)$–dimensional matrix equation in the space of the quarks, antiquarks and gluons. This equation is a fundamental equation for the parton distribution functions as well as the parton shower.

When developing an event simulation, the PDFs will define which parton is drawn from the hadron for the interaction. The remaining part of the hadron is known as the beam remnant. Since the way that hadrons are formed and stay bound lies in the non-perturbative regime of QCD and is therefore not well understood, the only way to describe the remnants is through phenomenological models.

The beam remnant carries a portion of the beam hadron's momentum and often will travel straight down the beam pipe, leading to undetectable physics. This isn't always the case, however, and these remnants can lead to some detectable physical results and as such are worth studying. There are very simple models, such as the UA5 model [21] that can be used. This model is simply a parameterization of the data and does not scale to higher energies. These simple models lack the adequate ability to describe all the effects of the remnants. Instead there is evidence to suggest that these remnant can and do interact again within one event. These models are known as multiple interaction models [22,23] and are currently still an interesting point of research. The whole process of the beam remnants and their interactions is known as the *Underlying Event*.

1.3.4 The Parton Shower

The hard process will generate a set of final-state particles for a given set of incoming particles. As the hard process is only accurate up to a given order, it isn't able to describe events with high parton multiplicity. As colliders are able to achieve higher energies, these high multiplicity events begin to play a larger role in the events. There is a need to generate these higher multiplicity final states to some approximation. The *Parton Shower* is able to generate these states in the soft and collinear regions of phase space to all orders. It is also in these soft and collinear regions which the higher order matrix elements are enhanced.

In this section I explain the parton shower and how it describes the enhanced soft and collinear regions [4]. The parton shower is an approximate perturbative treatment of QCD at momentum transfer-squared t greater than some infra-red cutoff t_0. This treatment is then easy to integrate with a hadronization model which begins at the cutoff, t_0, and turns the partonic final state into hadronic states.

In order to generate the higher multiplicity states we must split a parton a into two partons $b + c$. We start by assuming

$$p_b^2, p_c^2 \ll p_a^2 \equiv t. \tag{1.80}$$

If we have a as an outgoing parton, this corresponds to a *timelike* shower ($t > 0$). If it is an incoming parton, this is a *spacelike* shower. For the following introduction we will consider the timelike shower. It can be shown that the spacelike shower retains the same formulation, it just requires different kinematics.

We can define the energy fraction as

$$z = E_b/E_a = 1 - E_c/E_a. \tag{1.81}$$

Now in the region of small angles, which is the region where the matrix element is enhanced, we have

$$t = 2E_bE_c(1 - \cos\theta) = z(1 - z)E_a^2\theta^2, \tag{1.82}$$

and using transverse momentum conservation

$$\theta = \frac{\theta_b}{1 - z} = \frac{\theta_c}{z}. \tag{1.83}$$

We then find that the matrix element squared for $n+1$ partons, from a $g \to gg$ splitting in the small angle approximation, is

$$|\mathcal{M}_{n+1}|^2 \sim \frac{4g^2}{t} C_A F(z; \epsilon_a, \epsilon_b, \epsilon_c) \left|\mathcal{M}_n\right|^2, \tag{1.84}$$

where $C_A = 3$ comes from $f^{ABC}f^{ABC}$ and the function F is the helicity dependent splittings. If we average this over the incoming gluon spins and sum it over the outgoing

gluon spins we get

$$C_A \langle F \rangle \equiv \hat{P}_{gg}(z) = C_A \left[\frac{1-z}{z} + \frac{z}{1-z} + z(1-z) \right]. \tag{1.85}$$

This is the unregularized *gluon splitting function* related to the Altarelli-Parisi kernel [3]. If we do the same thing for $g \to q\bar{q}$ and $q \to qg$ we find the unregularized splitting functions

$$\hat{P}_{qg}(z) = T_R \left[z^2 + (1-z)^2 \right], \tag{1.86}$$

$$\hat{P}_{qq}(z) = C_F \frac{1+z^2}{1-z}. \tag{1.87}$$

Here $T_R = \mathrm{Tr}(t^A t^A)/8 = 1/2$ and $C_F = \mathrm{Tr}(t^A t^A)/3 = 4/3$. Therefore we generally find

$$|\mathcal{M}_{n+1}|^2 \sim \frac{4g^2}{t} \hat{P}_{ba} |\mathcal{M}_n|^2 . \tag{1.88}$$

We then find the new differential cross section for the $n+1$ state is

$$d\sigma_{n+1} = d\sigma_n \frac{dt}{t} dz \frac{\alpha_{\mathrm{S}}}{2\pi} \hat{P}_{ba}(z), \tag{1.89}$$

where $\hat{P}_{ba}(z)$ is the appropriate splitting function to generate the new state.

The parton shower doesn't just emit one gluon in the soft or collinear region. It is capable of multiple branchings. We start by introducing the *Sudakov form factor*

$$\Delta(t) \equiv \exp \left[- \int_{t_0}^{t} \frac{dt'}{t'} \int dz \frac{\alpha_s}{2\pi} \hat{P}(z) \right]. \tag{1.90}$$

Doing this we can write the DGLAP evolution equation (1.79) as

$$t \frac{\partial}{\partial t} f(x,t) = \int_x^1 \frac{dz}{z} \frac{\alpha_{\mathrm{S}}}{2\pi} \hat{P}(z) f(x/z,t) + \frac{f(x,t)}{\Delta(t)} t \frac{\partial}{\partial t} \Delta(t), \tag{1.91}$$

which is equivalent to

$$t\frac{\partial}{\partial t}\left(\frac{f}{\Delta}\right) = \frac{1}{\Delta}\int_x^1 \frac{dz}{z}\frac{\alpha_{\rm S}}{2\pi}\hat{P}(z)f(x/z,t). \tag{1.92}$$

This has the solution

$$f(x,t) = \Delta(t)f(x,t_0) + \int_{t_0}^t \frac{dt'}{t'}\frac{\Delta(t)}{\Delta(t')}\int_x^1 \frac{dz}{z}\frac{\alpha_{\rm S}}{2\pi}\hat{P}(z)f(x/z,t'). \tag{1.93}$$

From this equation, the Sudakov form factor is interpreted as the probability of evolving from t_0 to t without branching. The infra-red singularity of the splitting function is still not handled, however, so we must impose an upper limit on z such that $z < 1-\epsilon(t)$. The branching for values of z above this are interpreted as *unresolvable*. These are emissions of gluons that are so soft they are undetectable. With the cutoff, the form factor is interpreted as the probability of not having any *resolvable* branchings from t_0 to t.

This idea of the Sudakov form factors easily integrates into the Monte Carlo method. From the interpretation of the form factors we find that the probability of evolving from t_1 to t_2 without (resolvable) branching is $\Delta(t_2)/\Delta(t_1)$. For time-like branchings we can find the distibution of t_2 by solving

$$\frac{\Delta(t_1)}{\Delta(t_2)} = \mathcal{R}. \tag{1.94}$$

where $\mathcal{R}$ is a random number in the interval $[0,1]$.

When evolving space-like partons the structure of the hadron must be maintained. We can see from (1.92) that this requires using the ratio $\Delta(t_i)/f(x,t_i)$. This means we must find the distribution of t_2 by solving

$$\frac{f(x,t_1)\Delta(t_2)}{f(x,t_2)\Delta(t_1)} = \mathcal{R}, \tag{1.95}$$

In either case if the value of t_2 is higher than the hard subprocess, Q^2, then we have reached the no branching condition for that parton.

In the timelike case, if a branching does occur, we then need to calculate the momentum fraction, z. This is done by solving

$$\int_{\epsilon}^{x_2/x_1} dz \frac{\alpha_s}{2\pi} P(z) = \mathcal{R}' \int_{\epsilon}^{1-\epsilon} dz \frac{\alpha_s}{2\pi} P(z), \tag{1.96}$$

where $\mathcal{R}'$ is another random number in the interval $[0, 1]$. To construct the momentum of the products of the emission we simply need to generate an azimuthal angle in the interval $[0, 2\pi]$. If polarization correlations are taken into account, then this angle won't be uniform.

The spacelike case isn't very different. In this case we are evolving backwards and trying to maintain the structure of the beam particle which is known from the PDF. This requires the modification to the distribution of z of

$$\int_{\epsilon}^{x_2/x_1} dz \frac{\alpha_s}{2\pi} \frac{P(z)}{z} f(x_2/z, t_1) = \mathcal{R}' \int_{\epsilon}^{1-\epsilon} dz \frac{\alpha_s}{2\pi} \frac{P(z)}{z} f(x_2/z, t_1). \tag{1.97}$$

Another important aspect of these branchings is angular ordering [4]. *Coherent parton branching* shows that each successive emission must lie within a cone with half angle given by the previous emission. This effect is also present in QED radiation. In this case it is easier to explain the angular ordering effect. Since the charge partner is created with angle, θ, an emission outside of this angle would effectly appear to be an emission from a chargeless object. In a sense, it can't be determined which particle this emission came from, making it an incoherent emission. Therefore a coherent emission from a particle must be emitted within the half angle of the previous emission. A similar logic can be applied to QCD radiation, though it is not as straightforward due to the three degrees of colour charge but yields the same result. This means that each emission

is restricted to have a smaller angle then the previous emission.

In order to have a coherent parton branching, the shower needs to evolve with a variable related to θ, rather then one related to the virtuality. This is given by

$$\zeta = \frac{p_b \cdot p_c}{E_b E_c} \simeq 1 - \cos\theta. \tag{1.98}$$

In HERWIG [24] the variable used, which ensures angular ordering, is

$$\tilde{t} = E^2 \zeta \geq t_0. \tag{1.99}$$

In Pythia [25] an angular ordered variable is not used. Instead, a virtuality ordered shower is used and emissions that violate angular ordering are vetoed. Part of my research has been into a new set of evolution variables. This new work is discussed in Chapter 2.

The shower does have some more complications, however. It is possible to calculate matrix elements to higher order. It is desirable to use these matrix elements as they contain all of the terms at a particular order, whereas the shower includes some of the terms at next-to-leading log (NLL). As these higher order matrix elements are enhanced in the soft and collinear regions it is desirable to use the shower in these regions and the matrix element in the hard regions. In order to do so properly, over- and under-counting must be prevented. This process is called the *matrix element corrections*.

There are two types of corrections, hard and soft. The soft corrections are when the shower approximation is corrected so that it is closer to the hard matrix element. This means that a gluon from the shower is prevented from occuring as it has already been properly considered in the matrix element. The hard corrections are when the first gluon emitted is determined to come from the matrix element, rather than the shower. The region where the matrix element is used and the region where the shower is used

need to match smoothly in order to correctly describe the physics and these corrections ensure the smooth matching of the two.

The matching of the parton shower and the higher order matrix elements is important for generating useful results and ensuring that the simulations describe the theory as accurately as it can. There are two different problems with matching higher order diagrams. The first is in ensuring that all of phase space is properly covered by the next-to-leading order matrix element [26]. There is also a need to ensure the shower and matrix element match at next-to-leading log (NLL) without double counting [27]. The decays of heavy quarks and other heavy coloured particles (such as SUSY particles) can also involve the parton shower. The showering of these processes and the matrix element corrections to them have also been studied [28].

1.3.5 Hadronization

After the parton shower, we are left with a set of partons that are of the same order in virtuality as the cutoff on the parton shower, t_0. At this stage the interactions between the partons become heavily influenced by non-perturbative effects in the low momentum-transfer, long distance regime.

A hadronization model must take the partons from virtuality t_0 down into stable and unstable hadrons. We would expect that there would be more hadrons created when t_0 is larger, as they have a higher virtuality. Likewise we would expect there to be more partons at the initial stage of hadronization if t_0 is smaller.

Since hadronization models aren't that well understood, it turns out that carefully letting the parton shower run to lower scales produces better results. This implies that though the parton shower only contains perturbative results, these are often still more valid than hadronization models. Quarks and gluons are not observed in collider

experiments, hadrons are; therefore in the end we must use a hadronization model to study collider physics.

The simplest model is to assume that each parton produces hadrons independently of the other partons. This is known as the *Independent Fragmentation Model*. This was originally designed by Field and Feynman [29] to approximate scaling of energy distributions observed in quark jets in e^+e^- events at moderate energy. This model takes a quark and pairs it with an anti-quark in a $q\bar{q}$ pair drawn from the vacuum. This forms a meson and the remaining quark fragments the same way. This continues until the leftover energy falls below a threshold. In this model the gluon is split into $q\bar{q}$ pairs. The momentum can be distributed in many ways. All the momentum can be given to one of the pair so a gluon behaves just like a quark in this model. The momentum could also be given by the $g \to q\bar{q}$ Altarelli-Parisi splitting function. This model has several problems. As the final partons are supposed to be on mass-shell it can lead to momentum conservation problems. Since there are low energy quarks remaining in this model, there is also a problem of colour flow.

Another model is the *String Model*. This model assumes that two colour connected partons have some colour field between them that grows with seperation. It is usually assumed to have a uniform energy per length. This amounts to a linear quark confining potential. When this "string" between the two quarks contains too much energy it breaks and a $q\bar{q}$ pair fill each side of the break. This continues until each string is considered "stable". At this point each part forms hadrons from the flavours that are colour connected. Gluons in this model form kinks in the strings because they carry a localized energy and momentum. It is the hadronization of these kinked strings that generates results that match experiments better than the independent fragmentation model. This is the model that is implemented in Pythia [25].

The model that is implemented in HERWIG is the the *cluster hadronization model*.

This model tries to cluster quarks together to form hadrons. The cluster model of hadronization relies on the colour preconfinement property of parton branching [30]. This property implies that pairs of colour-connected neighbouring partons have a mass distribution that falls off rapidly at high masses. It then makes sense to cluster these partons into colour-singlet clusters that can decay into observable hadrons.

The gluons in a cluster model must be split in order to form clusters. This is done non-perturbatively as we are in the non-perturbative regime. This means that the gluons are just split as a two body decay. This allows the colour connected quarks to form colour singlet clusters. It is important to note that in order to do the non-perturbative splitting, the gluons have to be given an effective mass. The value of this mass dictates the available flavours for the splitting. A common value of 750 MeV is used and this allows the gluon to split into u and d flavours. A new cluster model [31] is also used in the new event generator SHERPA [32].

I have developed a new model based on the old HERWIG cluster model. This new model and the improvements it provides are discussed in Chapter 3.

1.3.6 Decays

The last part to using the Monte Carlo method in collider simulations is the decays of the hadrons. There are thousands of decay modes of hadrons and unstable particles from the Particle Data Group [33]. Some of these modes are quite rare, others quite common. Since there are so many modes, and most of them are for hadrons, calculating a distribution for the decay particles is not easy, or even always possible.

Though exact calculations aren't always possible, there may be simple distributions that are better fits to reality based on certain properties, such as parity violation. In Herwig++ a few of these simple matrix elements have been included, in much the same

way that they were included in HERWIG. The two commonly used decay matrix elements are for three-body decays with the decaying particle of mass m_0 decaying into particles of masses m_1, m_2 and m_3. For free particles that decay weakly the free massive $(V-A)^2$ matrix elements is used. The decay momenta, q_i, are generated in the three body phase space with the weight $\mathcal{W}(m_0; m_1, m_2, m_3)$ given by

$$\mathcal{W}(m_0; m_1, m_2, m_3) = \frac{4(m_0^2 + m_1^2 - \mathcal{S})(\mathcal{S} - m_2^2 + m_3^2)}{(m_0^2 + m_1^2 - m_2^2 - m_3^2)^2}, \tag{1.100}$$

where $\mathcal{S}$ is $(q_2 + q_3)^2$ and is given by

$$\mathcal{S} = \mathcal{R}(m_2 + m_3)^2 + (1 - \mathcal{R})(m_0 - m_1)^2. \tag{1.101}$$

For bound particles that decay weakly this same matrix element is used but it has a veto placed on it. This veto is to ensure that the decay products are moving away from each other fast enough to no longer be bound.

Decays aren't always confined to happening after hadronization, however. In the SM, for example, the top quark will decay before it ever hadronizes. Also the τ will decay before it leaves the detector. In fact in SUSY there are many more particles that decay before they hadronize. Often these particles are fermions and so spin correlations can play an important part in the distributions. I present here a method which can be integrated into Monte Carlo simulations to provide the full spin correlations to these particles [34].

This algorithm is difficult to explain in a completely abstract manner. Instead an example will be presented here. Consider a $2 \to n$ hard subprocess. Here we label the incoming particles a_1 and a_2 and the outgoing particles b_1 to b_n. The momenta of these

particles are given by the matrix element

$$\rho^1_{\kappa_1\kappa_1'}\rho^2_{\kappa_2\kappa_2'}\mathcal{M}_{\kappa_1\kappa_2;\lambda_1\ldots\lambda_n}\mathcal{M}^*_{\kappa_1'\kappa_2';\lambda_1'\ldots\lambda_n'}\prod_{i=1,n}D^i_{\lambda_i\lambda_i'}, \tag{1.102}$$

where ρ is the spin density matrix of the incoming particles, κ_i are the incoming particles helicity, $\mathcal{M}$ is the matrix element of the $2\to n$ process, λ_i is the helicity of the b_i and $D^i_{\lambda_i\lambda_i'}$ is the decay matrix for the b_i. Initially all the D^i are just $\delta_{\lambda_i\lambda_i'}$ and the spin density matrices are given as $\rho_{\kappa_1\kappa_1'}=\delta_{\kappa_1\kappa_1'}$ for unpolarized incoming particles and

$$\rho_{\kappa_1\kappa_1'}=\begin{pmatrix}\frac{1}{2}(1+\mathcal{P}_3) & 0\\ 0 & \frac{1}{2}(1-\mathcal{P}_3)\end{pmatrix}, \tag{1.103}$$

for longitudinally polarized spin 1/2 incoming particles. Here $\mathcal{P}_3$ is the component of the polarization parallel to the beam axis.

Next a b_j is chosen at random. The spin density for this particle is given by

$$\rho_{\lambda_j\lambda_j'}=\frac{1}{N_\rho}\rho^1_{\kappa_1\kappa_1'}\rho^2_{\kappa_2\kappa_2'}\mathcal{M}_{\kappa_1\kappa_2;\lambda_1\ldots\lambda_n}\mathcal{M}^*_{\kappa_1'\kappa_2';\lambda_1'\ldots\lambda_n'}\prod_{i\neq j}D^i_{\lambda_i\lambda_i'}, \tag{1.104}$$

where the normalization N_ρ is chosen so the trace of the spin density matrix is one. The decay mode of this particle is selected based on the branching ratios. This produces particles c_1 to c_m with helicity υ_i. The momentum of these particles is given by the matrix element

$$\rho_{\lambda_j\lambda_j'}\mathcal{M}_{\lambda_j;\upsilon_1\ldots\upsilon_m}\mathcal{M}^*_{\lambda_j';\upsilon_1'\ldots\upsilon_m'}\prod_{i=1,m}D^i_{\upsilon_i,\upsilon_i'}. \tag{1.105}$$

Another randomly selected decay product, c_k, is chosen from the decay of b_j. The spin density for this new decay product is

$$\rho_{\upsilon_k,\upsilon_k'}=\frac{1}{N_{D\rho}}\rho_{\lambda_j\lambda_j'}\mathcal{M}_{\lambda_j;\upsilon_1\ldots\upsilon_m}\mathcal{M}^*_{\lambda_j;\upsilon_1\ldots\upsilon_m}\prod_{i\neq k}D^i_{\upsilon_i\upsilon_i'}, \tag{1.106}$$

where again the normalization is chosen so the trace of the spin density matrix is one.

This process of decaying the products continues all the way up the decay chain until a stable particle is reached. Once this occurs the decay matrix is fixed as an identity matrix (i.e. $\delta_{\lambda_i\lambda'_i}$). Once all of the decay products of a particle have been handled the decay matrix for the particle is calculated. Returning to our example, assume that the particle b_j has had all of its decay products generated. Its decay matrix would be

$$D_{\lambda_j\lambda'_j} = \frac{1}{N_D}\mathcal{M}_{\lambda_j;v_1\ldots v_m}\mathcal{M}^*_{\lambda'_j;v_1\ldots v'_j}\prod_{i=1,m} D^i_{v_i v'_i}, \tag{1.107}$$

where this too is normalized so the trace is one.

This whole process continues for all b_j until all the decay products have had their spin density matrices and there decay matrices calculated. Since all the spin information is passed up the chain via the spin density matrices and then passed back down the chain via the decay matrices, the whole event has the complete spin correlations built into the decay products. For a more thorough discussion and examples of the spin correlation effects see [34].

There are many packages available that perform selected decays using more advanced algorithms. These can almost always be implemented into the Monte Carlo simulations at this point. One example of this is EvtGen [35]. This package uses decay amplitudes, rather than probabilities, so it can correctly generate the angular correlations in a decay chain. Many of the decay modes in this package have been developed from experimental data and therefore, match data much better than the simple model built into Herwig++.

Chapter 2

New formalism for QCD parton showers

2.1 Introduction

The parton shower approximation has become an important component of a wide range of comparisons between theory and experiment in particle physics. Calculations of observables that are asymptotically insensitive to soft physics are known as *infrared safe* observables. These can be performed in fixed-order perturbation theory, but the resulting final states consist of a few isolated partons, quite different from the multihadron final states observed experimentally. One can attempt to identify isolated partons with hadronic jets, but then the energy flows within and between jets are not well represented.

Currently, the only means of connecting few-parton states with the real world is via parton showers, which generate high-multiplicity partonic final states in an approximation that retains enhanced collinear and soft contributions to all orders. Such multiparton states can be interfaced to a hadronization model which does not require large

momentum transfers in order to produce a realistic hadronic final state. Hadronization and detector corrections to the fixed-order predictions can then be computed, and the results have generally been found to be in satisfactory agreement with the data. Infrared-sensitive quantities such as hadron spectra and multiplicities have also been described successfully using parton showers. This has strengthened the belief that similar techniques can be used to predict new physics signals and backgrounds in future experiments.

This chapter presents a new shower evolution formalism [36], based on an angular variable related to transverse momentum [37–40]. The main aim of these new variables is to retain the direct angular ordering of the shower while improving the Lorentz invariance of the evolution and simplifying the coverage of phase space, especially in the soft region. The old shower variables used in HERWIG used massless splitting functions which created an artificial lower bound for the transverse momentum. This created an artificial "dead cone" in the emission from heavier quarks in which no emissions could lie. By allowing evolution down to zero transverse momentum and the use of mass-dependent splitting functions, the new shower variables permit a better treatment of heavy quark fragmentation which eliminates these the sharply-defined collinear "dead cones".

In the following section the new shower variables and their associated kinematics and dynamics are defined. The appropriate argument of the running coupling, the mass-dependent parton branching probability, and the shower evolution cutoff are also given. The variables are defined slightly differently for initial- and final-state parton branching, and depend on the colour connection of the evolving parton, so in subsequent sections the various possible configurations of colour flow between initial and final jets are considered.

The formalism presented here is implemented in the new Monte Carlo event generator Herwig++ [41] which is described in detail in Chapter 4. Results for e^+e^- annihilation

and comparisons with LEP data have been presented in a separate publication [42] and are also given in Chapter 5. The formulae in this chapter could also be used to construct a matching scheme for next-to-leading order (NLO) QCD calculations and Herwig++ parton showers, similar to that developed for HERWIG showers in [43, 44] and implemented in the MC@NLO event generator [45].

2.2 New variables for parton branching

As mentioned in Chapter 1 there are two types of shower evolutions. When a parton is space-like ($t < 0$) it is an initial-state parton and is described here as part of the initial-state shower. When a parton is time-like ($t > 0$) it is a final-state parton and is described as part of the final-state shower. Each individual splitting of a parton is referred to as a branching. The complete branching history of a given parton is called its evolution and the collection of all the evolutions of all the final- (initial-) state partons is referred to as the final- (initial-) state shower.

2.2.1 Final-state quark branching

2.2.1.1 Kinematics

Consider parton branching in an outgoing (heavy) quark jet. Define the quark momentum after the ith gluon emission $q_{i-1} \to q_i + k_i$ (see figure 2.1) in the *Sudakov basis* as

$$q_i = \alpha_i p + \beta_i n + q_{\perp i} \tag{2.1}$$

where p is the jet's "parent parton" momentum ($p^2 = m^2$, the *on-shell* quark mass-squared), n is a lightlike "backward" 4-vector ($n^2 = 0$), and $q_{\perp i}$ is the transverse mo-

mentum ($q_{\perp i}^2 = -\mathbf{q}_{\perp i}^{\,2}$, $q_{\perp i} \cdot p = q_{\perp i} \cdot n = 0$). Then

$$\beta_i = \frac{\mathbf{q}_{\perp i}^{\,2} + q_i^2 - \alpha_i^2 m^2}{2\alpha_i p \cdot n} \,. \tag{2.2}$$

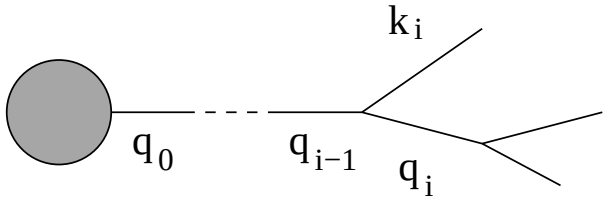


Figure 2.1: Final-state parton branching. The blob represents the hard subprocess.

The momentum fraction and relative transverse momentum are now defined as

$$z_i = \frac{\alpha_i}{\alpha_{i-1}} \,, \qquad \mathbf{p}_{\perp i} = \mathbf{q}_{\perp i} - z_i \mathbf{q}_{\perp i-1} \,. \tag{2.3}$$

Then we have

$$q_{i-1}^2 = \frac{q_i^2}{z_i} + \frac{k_i^2}{1-z_i} + \frac{\mathbf{p}_{\perp i}^{\,2}}{z_i(1-z_i)} \,. \tag{2.4}$$

2.2.1.2 Running coupling

As mentioned in Chapter 1 the QCD coupling constant, $\alpha_{\rm s}$ is scale dependent. This means that we need to decide what the right scale of a branching is. To find the optimal argument of $\alpha_{\rm s}$, we consider the branching of a quark of virtuality q^2 into an on-shell quark and an off-shell gluon of virtuality k^2 [46]. From (2.4), the propagator denominator is

$$q^2 - m^2 = \frac{1-z}{z} m^2 + \frac{k^2}{1-z} + \frac{\mathbf{p}_\perp^{\,2}}{z(1-z)} = \frac{1}{1-z} \left\{ k^2 + \frac{1}{z} [\mathbf{p}_\perp^{\,2} + (1-z)^2 m^2] \right\} \,. \tag{2.5}$$

The dispersion relation for the running coupling is supposed to be

$$\frac{\alpha_{\rm S}(\mu^2)}{\mu^2} = \frac{\alpha_{\rm S}(0)}{\mu^2} + \int_0^\infty \frac{dk^2}{k^2(k^2+\mu^2)}\rho_{\rm S}(k^2) \tag{2.6}$$

where $\rho_{\rm S}(k^2)$ is the discontinuity of $\alpha_{\rm S}(-k^2)$. The first term on the right-hand side comes from cutting through the on-shell gluon, the second from cutting through the gluon self-energy. In our case we have $k^2 + [\mathbf{p}_\perp{}^2 + (1-z)^2m^2]/z$ in place of $k^2+\mu^2$. We are interested in soft gluon resummation ($z \to 1$) [40] and so we ignore the factor of $1/z$ here. Thus the suggested argument of $\alpha_{\rm S}$ is $\mathbf{p}_\perp{}^2 + (1-z)^2m^2$. In practice a minimum virtuality is imposed on light quarks and gluons in the parton shower, and therefore the actual argument is slightly more complicated (see below).

2.2.1.3 Evolution variable

The evolution variable is not simply q^2 since this would ignore angular ordering. To have angular ordering, for massless parton branching, the evolution variable should be $\mathbf{p}_\perp{}^2/[z(1-z)]^2 = q^2/z(1-z)$ [47]. For gluon emission by a massive quark we assume this generalizes to $(q^2-m^2)/z(1-z)$. To define a resolvable emission we also need to introduce a minimum virtuality Q_g^2 for gluons and light quarks. Therefore from (2.5) the evolution variable is

$$\tilde{q}^2 = \frac{\mathbf{p}_\perp{}^2}{z^2(1-z)^2} + \frac{\mu^2}{z^2} + \frac{Q_g^2}{z(1-z)^2} \tag{2.7}$$

where $\mu = \max(m, Q_g)$. For the argument of the running coupling we use

$$z^2(1-z)^2\tilde{q}^2 = \mathbf{p}_\perp{}^2 + (1-z)^2\mu^2 + zQ_g^2\,. \tag{2.8}$$

The μ term allows for massive quarks to evolve down to $\mathbf{p}_\perp < (1-z)m$, i.e. inside the *dead cone* [48, 49].

Angular ordering of the branching $q_i \to q_{i+1}$ is defined by

$$\tilde{q}_{i+1} < z_i \tilde{q}_i \ . \tag{2.9}$$

The factor of z_i enters because the angle at each branching is inversely proportional to the momentum fraction of the parent. Similarly for branching on the gluon, $k_i \to k_{i+1}$, we require

$$\tilde{k}_{i+1} < (1-z_i)\tilde{q}_i \ . \tag{2.10}$$

2.2.1.4 Branching probability

For the parton branching probability we use the mass-dependent splitting functions given in ref. [39]. These are derived in the *quasi-collinear limit*, in which ${\mathbf{p}_\perp}^2$ and m^2 are treated as small (compared to $p \cdot n$) but ${\mathbf{p}_\perp}^2/m^2$ is not necessarily small. In this limit the $q \to qg$ splitting function is

$$P_{qq}(z, {\mathbf{p}_\perp}^2) = C_F \left[\frac{1+z^2}{1-z} - \frac{2z(1-z)m^2}{{\mathbf{p}_\perp}^2 + (1-z)^2 m^2} \right] \ . \tag{2.11}$$

It is the second term of this equation which differs from the splitting functions used in HERWIG. Note that at $\mathbf{p}_\perp = 0$ the factor in square brackets is just $1-z$, i.e. the soft singularity at $z \to 1$ becomes a zero in the collinear direction. The minimum virtuality Q_g^2 serves only to define a resolvable emission, and therefore we omit it when defining the branching probability in terms of the evolution variable (2.7) as

$$dP(q \to qg) = \frac{\alpha_{\rm S}}{2\pi} \frac{d\tilde{q}^2}{\tilde{q}^2} P_{qq}\, dz = \frac{C_F}{2\pi} \alpha_{\rm S}[z^2(1-z)^2\tilde{q}^2] \frac{d\tilde{q}^2}{\tilde{q}^2} \frac{dz}{1-z} \left[1 + z^2 - \frac{2m^2}{z\tilde{q}^2} \right] \ . \tag{2.12}$$

2.2.2 Gluon splitting

In the case of a final-state gluon splitting into a pair of heavy quarks of mass m, the quasi-collinear splitting function derived in [39] is

$$P_{qg}(z, {\mathbf{p}_\perp}^2) = T_R \left[1 - 2z(1-z)\frac{{\mathbf{p}_\perp}^2}{{\mathbf{p}_\perp}^2 + m^2}\right] . \tag{2.13}$$

We note that this splitting function is bounded above by its value $T_R = \frac{1}{2}$ at the phase space boundary $\mathbf{p}_\perp = \mathbf{0}$, and below by $T_R/2$. By analogy with (2.7), in this case the evolution variable $\tilde{q}$ is related to the virtuality of the gluon or the relative transverse momentum of the splitting by

$$\tilde{q}^2 = \frac{q^2}{z(1-z)} = \frac{{\mathbf{p}_\perp}^2 + m^2}{z^2(1-z)^2} . \tag{2.14}$$

In terms of the variables $\tilde{q}, z$, the $g \to q\bar{q}$ branching probability then reads

$$dP(g \to q\bar{q}) = \frac{T_R}{2\pi}\alpha_{\mathrm{S}}[z^2(1-z)^2\tilde{q}^2]\frac{d\tilde{q}^2}{\tilde{q}^2}\left[1 - 2z(1-z) + \frac{2m^2}{z(1-z)\tilde{q}^2}\right] dz . \tag{2.15}$$

In the case of gluon splitting into gluons, the branching probability takes the familiar form

$$dP(g \to gg) = \frac{C_A}{2\pi}\alpha_{\mathrm{S}}[z^2(1-z)^2\tilde{q}^2]\frac{d\tilde{q}^2}{\tilde{q}^2}\left[\frac{z}{1-z} + \frac{1-z}{z} + z(1-z)\right] dz . \tag{2.16}$$

Since we introduce a minimum virtuality Q_g^2 for gluons, the relationship between the evolution variable and the relative transverse momentum for this splitting is as in (2.14) but with the heavy quark mass m replaced by Q_g. Similarly, for gluon splitting to light quarks we use (2.14) with $\mu = \max(m, Q_g)$ in place of m.

2.2.3 Initial-state branching

Consider the initial-state (spacelike) branching of a partonic constituent of an incoming hadron that undergoes some hard collisions subprocess such as deep inelastic lepton scattering. The momenta are defined as in (2.1), with the reference vector p along the beam direction. In this case the evolution is performed *backwards* from the hard subprocess to the incoming hadron, as shown in figure 2.2. Thus we now define in place of

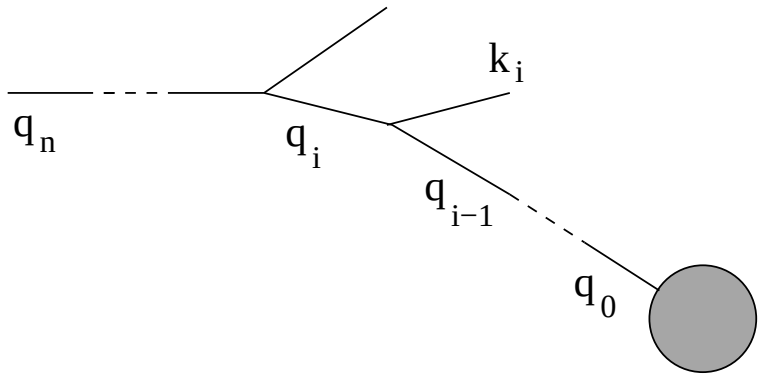


Figure 2.2: Initial-state parton branching. The blob represents the hard subprocess.

(2.3)

$$z_i = \frac{\alpha_{i-1}}{\alpha_i} , \qquad \mathbf{p}_{\perp i} = \mathbf{q}_{\perp i-1} - z_i \mathbf{q}_{\perp i} . \tag{2.17}$$

Then

$$q_{i-1}^2 = z_i q_i^2 - \frac{z_i}{1-z_i} k_i^2 - \frac{{\mathbf{p}_{\perp i}}^2}{1-z_i} . \tag{2.18}$$

We assume a massless variable-flavour-number evolution scheme [50, 51] for constituent parton branching, setting $m = 0$ and putting all emitted gluons at the minimum virtuality, $k_i^2 = Q_g^2$. The angular evolution variable now relates only to the angle of the emitted gluon and therefore we choose

$$\tilde{q}_i^2 = \frac{{\mathbf{p}_{\perp i}}^2 + z_i Q_g^2}{(1-z_i)^2} , \tag{2.19}$$

with ordering condition simply $\tilde{q}_{i+1} < \tilde{q}_i$. Correspondingly, for the argument of the running coupling we now use $(1-z)^2\tilde{q}^2$.

A different type of initial-state branching occurs in the decay of heavy, quasi-stable coloured objects like the top quark. Here the momentum of the incoming heavy object is fixed and evolution is performed forwards to the hard decay process. In this case we cannot neglect the mass of the parton and (2.19) becomes

$$\tilde{q}_i^2 = \frac{{\mathbf{p}_{\perp i}}^2 + z_i Q_g^2}{(1-z_i)^2} + m^2 \,, \tag{2.20}$$

while the branching probability (2.12) is replaced by

$$dP(q \to qg) = \frac{C_F}{2\pi}\alpha_{\rm S}[(1-z)^2\tilde{q}^2]\frac{d\tilde{q}^2}{\tilde{q}^2}\frac{dz}{1-z}\left[1+z^2-\frac{2zm^2}{\tilde{q}^2}\right] \,. \tag{2.21}$$

2.2.4 Allowed regions and termination of branching

The allowed phase space for each branching is given by requiring a real relative transverse momentum, ${\mathbf{p}_\perp}^2 > 0$. In final-state $q \to qg$ branching, we have from (2.8)

$$z^2(1-z)^2\tilde{q}^2 > (1-z)^2\mu^2 + zQ_g^2 \,. \tag{2.22}$$

This yields a rather complicated boundary in the $(\tilde{q}, z)$ plane. However, since

$$(1-z)^2\mu^2 + zQ_g^2 > (1-z)^2\mu^2 \,,\; z^2Q_g^2 \tag{2.23}$$

we see that the phase space lies inside the region

$$\frac{m}{\tilde{q}} < z < 1 - \frac{Q_g}{\tilde{q}} \,, \tag{2.24}$$

and approaches these limits for large values of $\tilde{q}$. The precise phase space can therefore be filled efficiently by generating values of z between these limits and rejecting those that violate the inequality (2.22). The resulting threshold for $\tilde{q}$ is slightly larger than but of the order of $m + Q_g$.

In gluon splitting, we obtain the allowed phase space range from (2.14) as

$$z_- < z < z_+, \qquad z_\pm = \frac{1}{2}\left(1 \pm \sqrt{1 - \frac{4\mu}{\tilde{q}}}\right) \qquad \text{and} \quad \tilde{q} > 4\mu \tag{2.25}$$

where $\mu = m$ for splitting into heavy quarks, or $\mu = \max(m, Q_g)$ more generally. Therefore, analogously to (2.24), the phase space lies within the range

$$\frac{\mu}{\tilde{q}} < z < 1 - \frac{\mu}{\tilde{q}}\,. \tag{2.26}$$

Schematically, the parton shower corresponds to selecting a sequence of $(\tilde{q}_i, z_i)$ values by solving the equations

$$\begin{aligned} \mathcal{R}_1 &= \exp\left(-\int_{\tilde{q}_i}^{\tilde{q}_{i-1}} d\tilde{q} \int_{z_-}^{z_+} dz\, \frac{d^2P}{d\tilde{q}\, dz}\right) \\ \mathcal{R}_2 &= \int_{z_-}^{z_i} dz\, \frac{d^2P}{d\tilde{q}\, dz} \Bigg/ \int_{z_-}^{z_+} dz\, \frac{d^2P}{d\tilde{q}\, dz} \end{aligned} \tag{2.27}$$

where $\mathcal{R}_{1,2} \in [0, 1]$ are uniform pseudorandom numbers. Whenever the algorithm selects a value of $\tilde{q}$ below the threshold, branching of that parton is terminated. The minimum virtuality Q_g thus determines the scale at which soft or collinear parton emission becomes unresolvable. In the absence of such a scale one eventually reaches a region where the perturbative expression for the running coupling is divergent.

One may wish to use a parameterization of α_s at low scales such that $\alpha_s(0)$ is finite. However, a cutoff Q_g is still needed to avoid divergence of the $q \to qg$ and $g \to gg$

branching probabilities. Alternatively one could consider parameterizing $\alpha_{\rm S}$ such that $\alpha_{\rm S}(0) = 0$, e.g.

$$\alpha_{\rm S}(q^2) = \frac{q}{Q_c}\alpha_{\rm S}(Q_c^2) \quad \text{for } q < Q_c \,, \tag{2.28}$$

where $Q_c > \Lambda$. Then the total branching probability below Q_c is (for massless quarks)

$$P_c(q \to qg) = C_F \frac{\alpha_{\rm S}(Q_c^2)}{2\pi} \int_0^1 2z(1+z^2)\, dz = \frac{\alpha_{\rm S}(Q_c^2)}{\pi} \,, \tag{2.29}$$

and no explicit cutoff is required, although of course Q_c is essentially playing the same rôle.

After branching has terminated, the outgoing partons are put on mass-shell (or given the virtual mass Q_g if lighter) and the relative transverse momenta of the branchings in the shower are computed. For final-state gluon splitting we have

$$|\mathbf{p}_\perp| = \sqrt{z^2(1-z)^2\tilde{q}^2 - \mu^2}, \tag{2.30}$$

or else, if the parent is a quark,

$$|\mathbf{p}_\perp| = \sqrt{(1-z)^2(z^2\tilde{q}^2 - \mu^2) - zQ_g^2}. \tag{2.31}$$

The virtualities of the internal lines of the shower can now be computed backwards according to (2.4). Finally, the azimuthal directions of the $\mathbf{p}_\perp$'s can be chosen [34] and the full 4-momenta reconstructed using eqs. (2.1) and (2.2).

In initial-state constituent parton branching the evolution is "guided" by the parton distribution functions (PDFs) of the incoming parent hadron. Since PDFs are often not tabulated below some scale $Q_s > Q_0$, one may wish to terminate branching whenever $\tilde{q} < Q_s$ is selected. In that case the incoming parton is assigned virtuality $q_n^2 \sim -Q_s^2$ and the spacelike virtualities of internal lines are then reconstructed back from q_n^2 to q_0^2

using the transverse momenta deduced from (2.19) inserted in (2.18).

For initial-state branching in the decay of a heavy, quasi-stable coloured object, the branching proceeds in the opposite direction but the reconstruction of momenta is similar, using (2.20) instead of (2.19).

2.2.5 Treatment of colour flows

The remaining sections of this chapter present a detailed treatment of the colour flows which depend on the choice of the "backward" vector n and on which quantities are to be held fixed during jet evolution. Normally n should be taken along the colour-connected partner of the radiating parton, and the 4-momentum of the colour-connected system should be preserved. The upper limits on the evolution variable $\tilde{q}$ for the colour-connected jets should be chosen so as to cover the phase space in the soft limit, with the best possible approximation to the correct angular distribution. In setting these limits we neglect the minimum virtuality Q_g^2, which is a good approximation at high energies. In the remaining sections we consider separately the four cases that the colour connection is between two final-state jets, two initial-state (beam) jets, a beam jet and a final-state jet, or a decaying heavy parton and a decay-product jet.

2.3 Final-final colour connection

Consider the process $a \to b + c$ where a is a colour singlet and b and c are colour-connected. Examples are $e^+e^- \to q\bar{q}$ and $W \to q\bar{q}'$. We need to preserve the 4-momentum of a and therefore we work in its rest-frame,

$$p_a = Q(1, \mathbf{0}, 0) \ , \quad p_b = \frac{1}{2}Q(1 + b - c, \mathbf{0}, \lambda) \ , \quad p_c = \frac{1}{2}Q(1 - b + c, \mathbf{0}, -\lambda) \ , \qquad (2.32)$$

where $p_a^2 = Q^2$, $b = m_b^2/Q^2$, $c = m_c^2/Q^2$ and

$$\lambda = \lambda(1, b, c) \equiv \sqrt{(1 + b - c)^2 - 4b} = \sqrt{(1 - b + c)^2 - 4c} \, . \tag{2.33}$$

For emission of a gluon g from b we write

$$q_i = \alpha_i p_b + \beta_i n + q_{\perp i} \tag{2.34}$$

where $\mathbf{q}_{\perp g} = \mathbf{k}_\perp$, $\mathbf{q}_{\perp b} = -\mathbf{k}_\perp$, $\mathbf{q}_{\perp c} = \mathbf{0}$ and we choose

$$n = \frac{1}{2} Q(\lambda, \mathbf{0}, -\lambda) \, . \tag{2.35}$$

Notice that, if c is massive, the alignment of n along p_c is exact only in a certain class of Lorentz frames. However, if we try to use a massive "backward" vector the kinematics become too complicated.

To preserve $p_a = q_b + q_c + q_g$ we require

$$\sum \alpha_i = \sum \beta_i = \frac{2}{1 + b - c + \lambda} \tag{2.36}$$

whereas the mass-shell conditions give

$$\begin{aligned} \beta_b &= \frac{2}{\lambda(1 + b - c + \lambda)} \left(\frac{b + \kappa}{\alpha_b} - b\alpha_b \right) \\ \beta_c &= \frac{2}{\lambda(1 + b - c + \lambda)} \left(\frac{c}{\alpha_c} - b\alpha_c \right) \\ \beta_g &= \frac{2}{\lambda(1 + b - c + \lambda)} \left(\frac{\kappa}{\alpha_g} - b\alpha_g \right) \end{aligned} \tag{2.37}$$

where $\kappa \equiv {\mathbf{k}_\perp}^2/Q^2$. Our new variables are

$$z = \frac{\alpha_b}{\alpha_b + \alpha_g} \, , \quad \tilde{\kappa} \equiv \frac{\tilde{q}^2}{Q^2} > b \, , \tag{2.38}$$

where from (2.7) we have

$$\kappa = (z^2\tilde{\kappa} - b)(1-z)^2 \ , \tag{2.39}$$

and so $\sqrt{b/\tilde{\kappa}} < z < 1$. From eqs.(2.36)-(2.39) we find

$$\begin{aligned} \alpha_b &= \frac{z}{1+b-c+\lambda}\left(1+b-c+z(1-z)\tilde{\kappa}+\sqrt{[1-b+c-z(1-z)\tilde{\kappa}]^2-4b}\right) \ , \\ \alpha_c &= \frac{2}{1+b-c+\lambda} - \frac{\alpha_b}{z} \ , \\ \alpha_g &= \frac{1-z}{z}\alpha_b \ , \end{aligned} \tag{2.40}$$

with the β_i's given by (2.37).

2.3.1 Phase space variables

It is convenient to express the phase space in terms of the Dalitz plot variables

$$x_i = \frac{2p_a \cdot q_i}{Q^2} = (1+b-c)\alpha_i + \lambda\beta_i \ . \tag{2.41}$$

Substituting from eqs. (2.37) and (2.40), we find

$$\begin{aligned} x_c &= 1-b+c-z(1-z)\tilde{\kappa} \\ x_b &= (2-x_c)r + (z-r)\sqrt{x_c^2-4c} \\ x_g &= (2-x_c)(1-r) - (z-r)\sqrt{x_c^2-4c} \end{aligned} \tag{2.42}$$

where

$$r = \frac{1}{2}\left(1+\frac{b}{1+c-x_c}\right) \ . \tag{2.43}$$

The Jacobian factor is thus simply

$$\frac{\partial(x_b,x_c)}{\partial(z,\tilde{\kappa})} = z(1-z)\sqrt{x_c^2-4c} \tag{2.44}$$

and the quasi-collinear branching probability (2.12) translates to

$$dP(q \to qg) = C_F \frac{\alpha_{\rm S}}{2\pi} \frac{dx_b\, dx_c}{(1-b+c-x_c)\sqrt{x_c^2-4c}} \left[\frac{1+z^2}{1-z} - \frac{2b}{1-b+c-x_c} \right] \tag{2.45}$$

where

$$z = r + \frac{x_b - (2-x_c)r}{\sqrt{x_c^2-4c}} \ , \tag{2.46}$$

r being the function of x_c given in (2.43).

For emission from parton c we write

$$q_i = \alpha_i p_c + \beta_i n + q_{\perp i} \tag{2.47}$$

where now we choose

$$n = \frac{1}{2} Q(\lambda, \mathbf{0}, \lambda) \ . \tag{2.48}$$

Clearly, the region covered and the branching probability will be as for emission from parton b, but with x_b and x_c, b and c interchanged.

2.3.2 Soft gluon region

For emission from parton b in the soft region $1 - z = \epsilon \to 0$ we have

$$x_c \sim 1 - b + c - \epsilon\tilde{\kappa} \ , \quad x_b \sim 1 + b - c - \epsilon\tilde{\kappa}' \tag{2.49}$$

where

$$\tilde{\kappa}' = \lambda + \frac{\tilde{\kappa}}{2b}(1 - b - c - \lambda) \ . \tag{2.50}$$

Since $\tilde{\kappa}$ is an angular variable, we can express it in terms of the angle θ_{bg} between the directions of the emitting parton b and the emitted gluon in the rest frame of a. In the

soft region we find

$$\tilde{\kappa} = \frac{(1+b-c+\lambda)(1+b-c-\lambda\cos\theta_{bg})}{2(1+\cos\theta_{bg})} \tag{2.51}$$

Thus $\tilde{\kappa} = b$ at $\theta_{bg} = 0$ and $\tilde{\kappa} \to \infty$ as $\theta_{bg} \to \pi$.

For soft emission from parton c, the roles of x_b and x_c, b and c are interchanged. To cover the whole angular region in the soft limit, we therefore require $\tilde{\kappa} < \tilde{\kappa}_b$ in jet b and $\tilde{\kappa} < \tilde{\kappa}_c$ in jet c. We also want the slope of the boundaries to match as they approach the soft limit, while still covering the divergence. In this case this gives the condition

$$\frac{\tilde{\kappa}_b}{\tilde{\kappa}'_b} = \frac{\tilde{\kappa}'_c}{\tilde{\kappa}_c} \tag{2.52}$$

and hence

$$(\tilde{\kappa}_b - b)(\tilde{\kappa}_c - c) = \frac{1}{4}(1-b-c+\lambda)^2 \ . \tag{2.53}$$

In particular, the most symmetric choice is

$$\tilde{\kappa}_b = \frac{1}{2}(1+b-c+\lambda) \ , \quad \tilde{\kappa}_c = \frac{1}{2}(1-b+c+\lambda) \ . \tag{2.54}$$

The largest region that can be covered by one jet corresponds to the maximal value of $\tilde{\kappa}$ allowed in (2.40) for real α_b, i.e. for the maximal b jet

$$\tilde{\kappa}_b = 4(1-2\sqrt{b}-b+c) \ . \tag{2.55}$$

2.3.3 Example: $e^+e^- \to q\bar{q}g$

Here we have $b = c = \rho$, $\lambda = \sqrt{1-4\rho} = v$, the quark velocity in the Born process $e^+e^- \to q\bar{q}$. The phase space and the two jet regions for the symmetrical choice (2.54) are

shown in figure 2.3. The region D, corresponding to hard non-collinear gluon emission, is not included in either jet and must be filled using the $\mathcal{O}(\alpha_s)$ matrix element (see below).

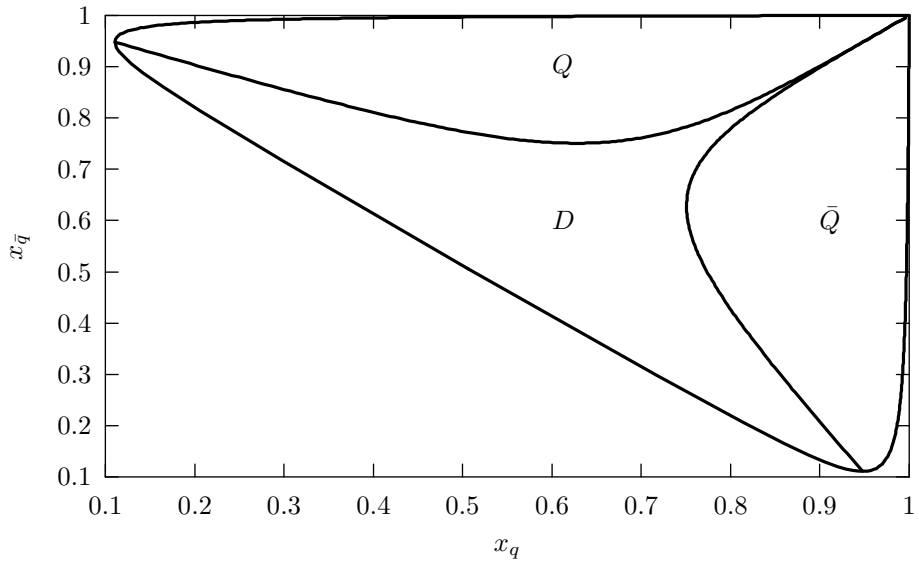


Figure 2.3: Phase space for $e^+e^- \to q\bar{q}g$ for $m_q = 5$ GeV, $Q^2 = m_Z^2$, with symmetric definition of quark and antiquark jets.

For the maximal quark jet we get from (2.55)

$$\tilde{\kappa}_q = 4(1 - 2\sqrt{\rho}) \ , \tag{2.56}$$

as shown in figure 2.4 together with the complementary antiquark jet region given by (2.53).

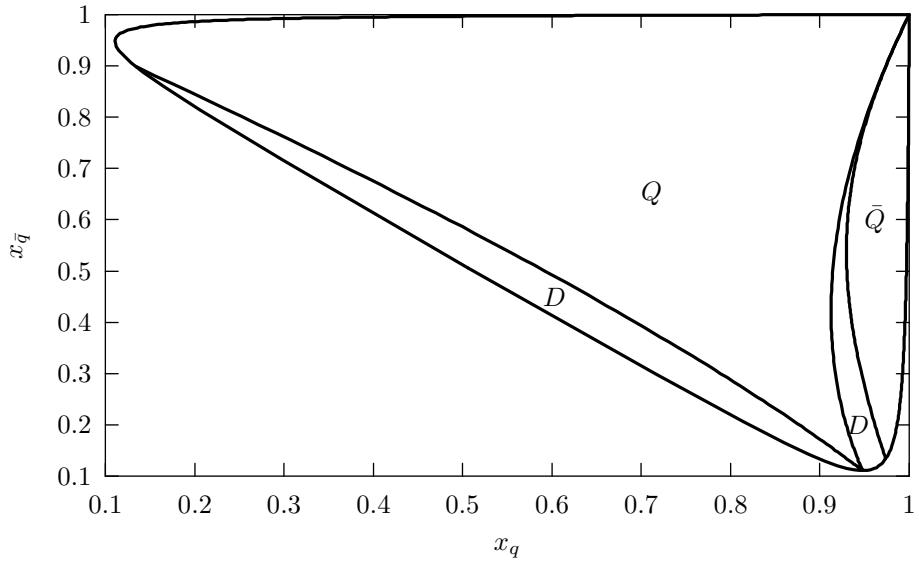


Figure 2.4: Phase space for $e^+e^- \to q\bar{q}g$ for $m_q = 5$ GeV, $Q^2 = m_Z^2$, with maximal region for the quark jet.

2.3.3.1 Exact matrix element

The $e^+e^- \to V \to q\bar{q}g$ differential cross section, where V represents a vector current such as a virtual photon, is given to first order in $\alpha_{\rm S}$ by [52, 53]

$$\frac{1}{\sigma_V}\frac{d^2\sigma_V}{dx_q\,dx_{\bar{q}}} = \frac{\alpha_{\rm S}}{2\pi}\frac{C_F}{v}\left[\frac{(x_q+2\rho)^2+(x_{\bar{q}}+2\rho)^2+\zeta_V}{(1+2\rho)(1-x_q)(1-x_{\bar{q}})} - \frac{2\rho}{(1-x_q)^2} - \frac{2\rho}{(1-x_{\bar{q}})^2}\right] \quad (2.57)$$

where

$$\zeta_V = -8\rho(1+2\rho) \quad (2.58)$$

and

$$\sigma_V = \sigma_0\,(1+2\rho)\,v \quad (2.59)$$

is the Born cross section for heavy quark production by a vector current, σ_0 being the massless quark Born cross section.

In the case of the axial current contribution $e^+e^- \to A \to q\bar{q}g$, instead of (2.57) we

have

$$\frac{1}{\sigma_A}\frac{d^2\sigma_A}{dx_q\,dx_{\bar{q}}} = \frac{\alpha_{\rm S}}{2\pi}\frac{C_F}{v}\left[\frac{(x_q+2\rho)^2+(x_{\bar{q}}+2\rho)^2+\zeta_A}{v^2(1-x_q)(1-x_{\bar{q}})} - \frac{2\rho}{(1-x_q)^2} - \frac{2\rho}{(1-x_{\bar{q}})^2}\right] , \quad (2.60)$$

where

$$\zeta_A = 2\rho[(3+x_g)^2 - 19 + 4\rho] , \quad (2.61)$$

σ_A being the Born cross section for heavy quark production by the axial current:

$$\sigma_A = \sigma_0 v^3 . \quad (2.62)$$

2.3.3.2 Soft gluon distribution

In the soft gluon region $1-z=\epsilon\to 0$ the branching probability (2.45) becomes

$$\begin{aligned}\frac{d^2P}{dx_q\,dx_{\bar{q}}} &\sim \frac{\alpha_{\rm S}}{2\pi}\frac{2C_F}{v\epsilon^2}f_s(\tilde{\kappa}) \\ f_s(\tilde{\kappa}) &= \frac{1}{\tilde{\kappa}} - \frac{\rho}{\tilde{\kappa}^2} . \end{aligned} \quad (2.63)$$

In this limit, the exact vector and axial current matrix elements, eqs. (2.57) and (2.60) respectively, give identical distributions:

$$\begin{aligned}\frac{1}{\sigma_V}\frac{d^2\sigma_V}{dx_q\,dx_{\bar{q}}} &\sim \frac{1}{\sigma_A}\frac{d^2\sigma_A}{dx_q\,dx_{\bar{q}}} \sim \frac{\alpha_{\rm S}}{2\pi}\frac{2C_F}{v\epsilon^2}f(\tilde{\kappa}) \\ f(\tilde{\kappa}) &= \frac{1-2\rho}{\tilde{\kappa}\tilde{\kappa}'} - \frac{\rho}{\tilde{\kappa}^2} - \frac{\rho}{\tilde{\kappa}'^2} \\ &= f_s(\tilde{\kappa})\left(\frac{v}{\tilde{\kappa}'}\right)^2 . \end{aligned} \quad (2.64)$$

Since from (2.50)

$$\tilde{\kappa}' = v + \tilde{\kappa}\left(\frac{1-v}{1+v}\right) > v , \quad (2.65)$$

the parton shower approximation (2.63) always overestimates the true result in the soft limit, and so correction by the rejection method is straightforward. For small values of ρ we have

$$f(\tilde{\kappa}) = \frac{1}{\tilde{\kappa}} - \frac{\rho}{\tilde{\kappa}^2} + \frac{2\rho^2}{\tilde{\kappa}} - 2\rho + \mathcal{O}(\rho^2) \; . \tag{2.66}$$

Since $\tilde{\kappa} > \rho$ we see that the error in the approximation (2.63) is at most $\mathcal{O}(\rho)$, for any value of $\tilde{\kappa}$ (figure 2.5).

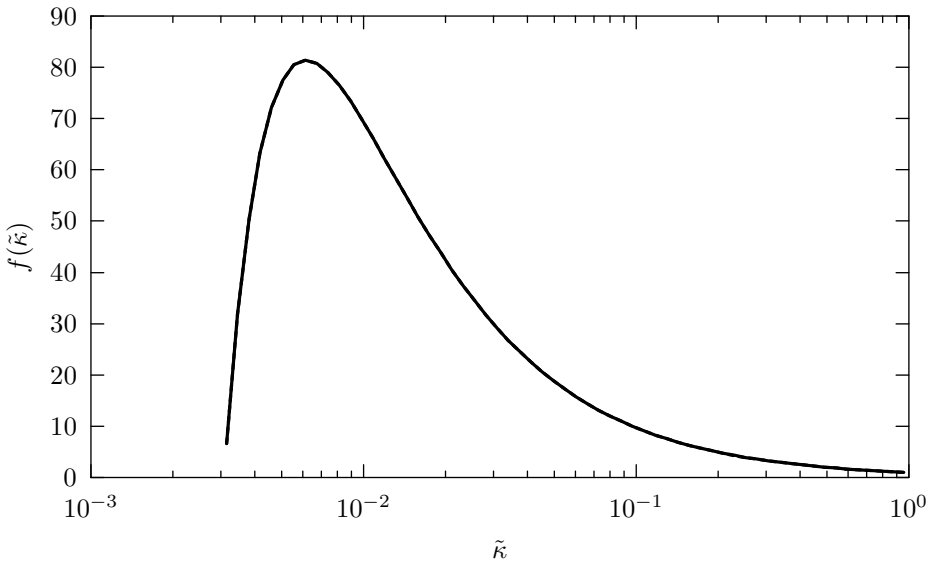


Figure 2.5: The function $f(\tilde{\kappa})$ giving the gluon angular distribution in the soft limit, for $m = 5$ GeV, $Q^2 = m_Z^2$. The exact result (2.64), solid curve, and shower approximation (2.63), dashed, are not distinguishable on this scale.

2.3.3.3 Dead region contribution

The integral over the dead region may be expressed as

$$\frac{1}{\sigma_V} \int_D d^2\sigma_V \equiv \frac{\alpha_{\rm S}}{2\pi} C_F \, F_V^D(\tilde{\kappa}_q) \tag{2.67}$$

where $\tilde{\kappa}_q$ parameterizes the boundary of the quark jet. As shown in figure 2.6, this is actually maximal, but still small, at the symmetric point given by (2.54).

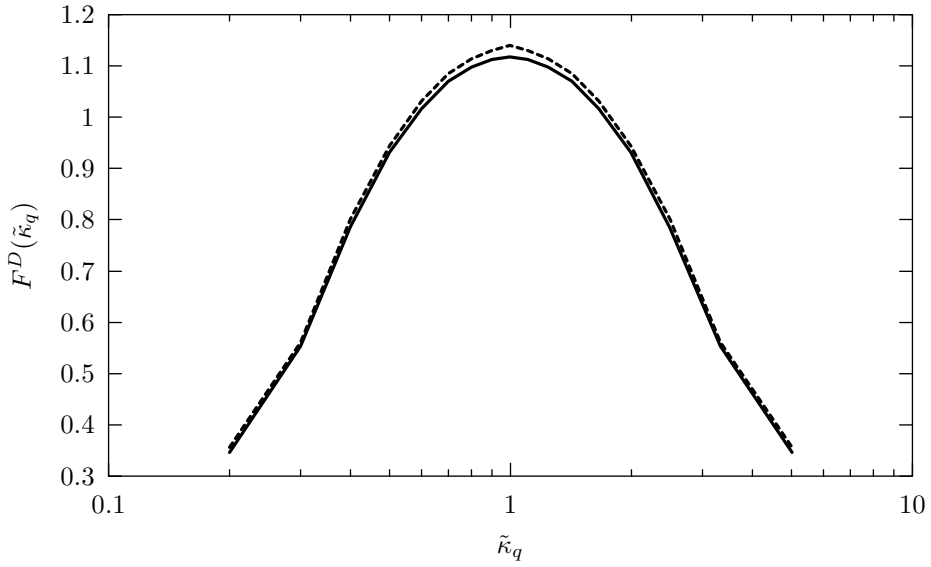


Figure 2.6: The function $F^D(\tilde{\kappa}_q)$ giving the contribution of the dead region to the cross section, for $m = 5$ GeV, $Q^2 = m_Z^2$. Solid: vector current. Dashed: axial current.

Although the integral in (2.67) is finite, the integrand diverges as one approaches the soft limit $x_q = x_{\bar{q}} = 1$ via the narrow "neck" of the dead region in figure 2.3 or 2.4. This could cause problems in generating $q\bar{q}g$ configurations in the dead region in order to apply a matrix element correction [54]. To avoid such problems, one can map the region $x_q, x_{\bar{q}} > \frac{3}{4}$ into a region whose width vanishes quadratically as $x_q, x_{\bar{q}} \to 1$, as illustrated in figure 2.7. The mapping shown is

$$\begin{aligned} x_q \to x'_q &= 1 - \left[\frac{1}{4} - (1 - x_q)\right] = \frac{7}{4} - x_q \,, \\ x_{\bar{q}} \to x'_{\bar{q}} &= 1 - 2(1 - x'_q)\left[\frac{3}{4} - (1 - x_q)\right] = \frac{5}{8} + \frac{1}{2}x_q + \frac{3}{2}x_{\bar{q}} - 2x_q x_{\bar{q}} \end{aligned} \tag{2.68}$$

when $x_q > x_{\bar{q}} > \frac{3}{4}$. Within the mapped region, the integrand then has an extra weight factor of $2(1 - x'_q)$ which regularizes the soft divergence. When $x_{\bar{q}} > x_q > \frac{3}{4}$, x_q and $x_{\bar{q}}$

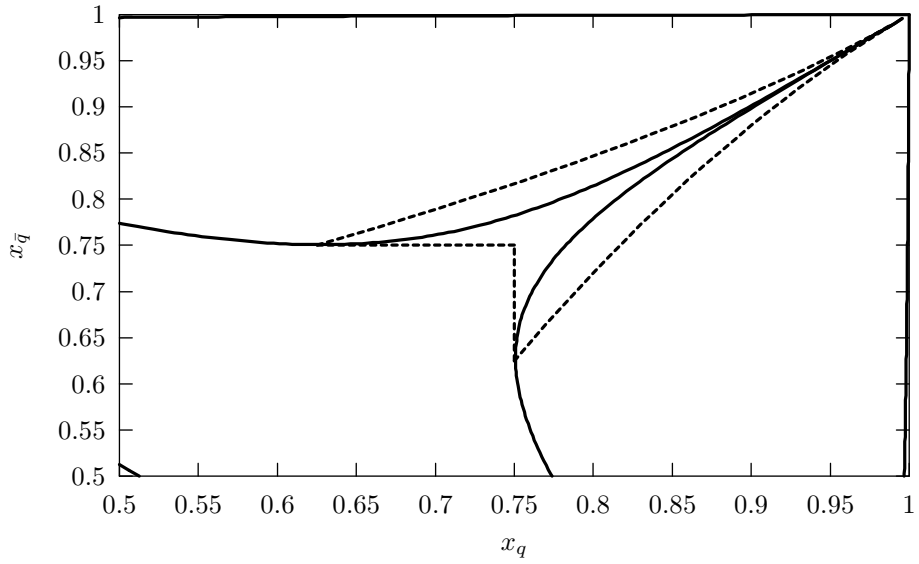


Figure 2.7: The soft region, with jet boundaries (solid) and mapped region (dashed), for $m = 5$ GeV, $Q^2 = m_Z^2$.

are interchanged in both the mapping and the weight.

2.4 Initial-initial colour connection

Here we consider the inverse process $b + c \to a$ where a is a colour singlet of invariant mass Q and b, c are beam jets. The kinematics are simple because we take beam jets to be massless: in the c.m. frame

$$p_a = Q(1, \mathbf{0}, 0) \ , \quad p_b = \frac{1}{2} Q(1, \mathbf{0}, 1) \ , \quad p_c = \frac{1}{2} Q(1, \mathbf{0}, -1) \ . \tag{2.69}$$

For emission of a gluon g from b we write

$$q_i = \alpha_i p_b + \beta_i p_c + q_{\perp i} \tag{2.70}$$

where $\mathbf{q}_{\perp g} = \mathbf{k}_\perp$, $\mathbf{q}_{\perp a} = -\mathbf{k}_\perp$, $\mathbf{q}_{\perp b} = \mathbf{q}_{\perp c} = \mathbf{0}$. Notice that in this case the recoil transverse momentum is taken by the colour singlet a so we cannot preserve its 4-momentum. We choose to preserve its mass and rapidity, so that

$$\alpha_a = \beta_a = \sqrt{1+\kappa}\,, \tag{2.71}$$

where as before $\kappa \equiv {\mathbf{k}_\perp}^2/Q^2$. Now we have

$$\begin{aligned} \beta_b &= \alpha_c = 0\,, \quad \alpha_g\beta_g = \kappa\,, \\ \alpha_a &= \alpha_b - \alpha_g\,, \quad \beta_a = \beta_c - \beta_g\,, \end{aligned} \tag{2.72}$$

and our new variables in this case are

$$z = 1 - \frac{\alpha_g}{\alpha_b}\,, \quad \tilde{\kappa} \equiv \frac{\tilde{q}^2}{Q^2} = \frac{\kappa}{(1-z)^2}\,. \tag{2.73}$$

Thus we find

$$\begin{aligned} \alpha_a &= \beta_a = \sqrt{1+(1-z)^2\tilde{\kappa}}\,, \\ \alpha_b &= \frac{1}{z}\sqrt{1+(1-z)^2\tilde{\kappa}}\,, \\ \beta_c &= \frac{1+(1-z)\tilde{\kappa}}{\sqrt{1+(1-z)^2\tilde{\kappa}}}\,. \end{aligned} \tag{2.74}$$

2.4.1 Phase space variables

It is convenient to express the kinematics in terms of the "reduced" Mandelstam invariants:

$$\bar{s} = (q_b+q_c)^2/Q^2\,, \quad \bar{t} = (q_b-q_g)^2/Q^2\,, \quad \bar{u} = (q_c-q_g)^2/Q^2\,. \tag{2.75}$$

The phase space limits are

$$1 < \bar{s} < S/Q^2 \ , \quad 1 - \bar{s} < \bar{t} < 0 \ , \quad \bar{u} = 1 - \bar{s} - \bar{t} \tag{2.76}$$

where S is the beam-beam c.m. energy squared. In terms of the shower variables for beam jet b, we have

$$\bar{s} = \alpha_b \beta_c = \frac{1}{z}[1 + (1-z)\tilde{\kappa}] \ , \quad \bar{t} = -\alpha_b \beta_g = -(1-z)\tilde{\kappa} \ , \quad \bar{u} = -(1-z)\bar{s} \ . \tag{2.77}$$

Thus curves of constant $\tilde{\kappa}$ in the $(\bar{s}, \bar{t})$ plane are given by

$$\bar{t} = \frac{\tilde{\kappa}(1-\bar{s})}{\tilde{\kappa} + \bar{s}} \tag{2.78}$$

and the Jacobian factor for conversion of the shower variables to the Mandelstam invariants is

$$\frac{\partial(\bar{s}, \bar{t})}{\partial(z, \tilde{\kappa})} = \frac{1-z}{z}\bar{s} \ . \tag{2.79}$$

For the other beam jet c we have $\bar{t} \leftrightarrow \bar{u}$ and thus

$$\bar{t} = \frac{\bar{s}(1-\bar{s})}{\tilde{\kappa} + \bar{s}} \ . \tag{2.80}$$

We see that in order for the jet regions to touch without overlapping in the soft limit $\bar{s} \to 1$, $\bar{t} \to 0$, we need $\tilde{\kappa} < \tilde{\kappa}_b$ in jet b and $\tilde{\kappa} < \tilde{\kappa}_c$ in jet c, where $\tilde{\kappa}_c = 1/\tilde{\kappa}_b$. The most symmetrical choice is $\tilde{\kappa}_c = \tilde{\kappa}_b = 1$, as shown in figure 2.8, but we can take $\tilde{\kappa}_b$ or $\tilde{\kappa}_c$ as large as we like.

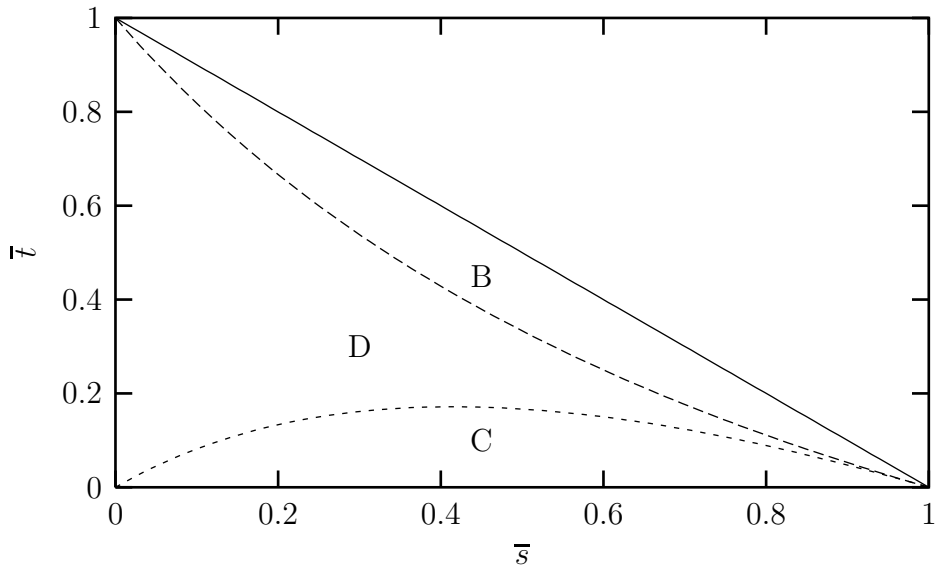


Figure 2.8: Beam jets (B,C) and dead region (D) in initial-state branching.

2.4.2 Example: Drell-Yan process

Consider radiation from the quark in the Drell-Yan process, $q\bar{q} \to gZ^0$. In the laboratory frame we have

$$q_q = (Px_1, \mathbf{0}, Px_1) \ , \quad q_{\bar{q}} = (Px_2, \mathbf{0}, -Px_2) \tag{2.81}$$

where $P = \frac{1}{2}\sqrt{S}$ is the beam momentum. If we generated the initial hard process $q\bar{q} \to Z^0$ with momentum fractions x_q, $x_{\bar{q}}$ and we want to preserve the mass and rapidity of the Z^0 we require

$$x_1 = x_q\alpha_b \ , \quad x_2 = x_{\bar{q}}\beta_c \tag{2.82}$$

where α_b and β_c are given by eqs (2.74).

The branching probability in the parton shower approximation is

$$\frac{d^2P}{dz\, d\tilde{\kappa}} = C_F \frac{\alpha_{\rm S}}{2\pi} \frac{1}{\tilde{\kappa}} \frac{1+z^2}{1-z} \ , \tag{2.83}$$

which gives a differential cross section ($s = \bar{s}Q^2$, etc.)

$$\frac{1}{\sigma_0}\frac{d^2\sigma}{ds\,dt} = \frac{D(x_1)D(x_2)}{D(x_q)D(x_{\bar{q}})}\frac{\alpha_{\rm S}}{2\pi}C_F\frac{s+u}{s^3tu}\left[s^2+(s+u)^2\right] \tag{2.84}$$

where σ_0 is the Born cross section. The functions $D(x_1)$ etc. are parton distribution functions in the incoming hadrons; these factors take account of the change of kinematics $x_q, x_{\bar{q}} \to x_1, x_2$ discussed above.

The exact differential cross section for $q\bar{q} \to gZ^0$ to order $\alpha_{\rm S}$ is

$$\frac{1}{\sigma_0}\frac{d^2\sigma}{ds\,dt} = \frac{D(x_1)D(x_2)}{D(x_q)D(x_{\bar{q}})}\frac{\alpha_{\rm S}}{2\pi}C_F\frac{Q^2}{s^3tu}\left[(s+t)^2+(s+u)^2\right]\ . \tag{2.85}$$

Since $Q^2 = s+t+u$ and $t \le 0$, we see that the parton shower approximation (2.84) overestimates the exact expression, becoming exact in the collinear or soft limit $t \to 0$. Therefore the gluon distribution in the jet regions can be corrected efficiently by the rejection method, and the dead region can be filled using the matrix element, as was done in [55]. The benefit of the new variables is that the angular distribution of soft gluon emission requires no correction, provided the jet regions touch without overlapping in the soft region. As shown above, this will be the case if the upper limits on $\tilde{\kappa}$ satisfy $\tilde{\kappa}_{\bar{q}} = 1/\tilde{\kappa}_q$.

2.5 Initial-final colour connection

Consider the process $a + b \to c$ where a is a colour singlet and the beam parton b and outgoing parton c are colour-connected. An example is deep inelastic scattering, where a is a (charged or neutral) virtual gauge boson. We need to preserve the 4-momentum

of a and therefore we work in the Breit frame:

$$p_a = Q(0, \mathbf{0}, -1) \ , \quad p_b = \frac{1}{2}Q(1+c, \mathbf{0}, 1+c) \ , \quad p_c = \frac{1}{2}Q(1+c, \mathbf{0}, -1+c) \ , \tag{2.86}$$

where $p_a^2 = -Q^2$, $p_b^2 = 0$, and $m_c^2 = cQ^2$. Notice that the beam parton b is always taken to be massless, but the outgoing parton c can be massive (e.g. in $W^+ d \to c$).

2.5.1 Initial-state branching

For emission of a gluon g from the incoming parton b we write

$$q_i = \alpha_i p_b + \beta_i n + q_{\perp i} \tag{2.87}$$

where $\mathbf{q}_{\perp g} = \mathbf{k}_\perp$, $\mathbf{q}_{\perp b} = \mathbf{0}$, $\mathbf{q}_{\perp c} = -\mathbf{k}_\perp$ and we choose

$$n = \frac{1}{2}Q(1+c, \mathbf{0}, -1-c) \ . \tag{2.88}$$

To preserve $p_a = q_c + q_g - q_b$ we now require

$$\alpha_b - \alpha_c - \alpha_g = \beta_c + \beta_g - \beta_b = \frac{1}{1+c} \tag{2.89}$$

whereas the mass-shell condition is

$$\alpha_i \beta_i \, Q^2 (1+c)^2 = {\mathbf{q}_{\perp i}}^2 + q_i^2 \tag{2.90}$$

which gives

$$1 + c = \frac{c}{\alpha_c} + \kappa \left(\frac{1}{\alpha_c} + \frac{1}{\alpha_g} \right) \ . \tag{2.91}$$

The new variables for emission from the beam jet are as in (2.73). Substituting in

(2.91), we find

$$\begin{aligned}
\alpha_b &= \frac{1}{2z(1+c)}\left(1+c+(1-z)\tilde{\kappa}+\sqrt{[1+c+(1-z)\tilde{\kappa}]^2-4z(1-z)\tilde{\kappa}}\right) , \\
\alpha_c &= z\alpha_b - \frac{1}{1+c} , \quad \alpha_g = (1-z)\alpha_b , \quad \beta_b = 0 , \\
\beta_c &= \frac{1}{1+c}\cdot\frac{c+(1-z)^2\tilde{\kappa}}{z(1+c)\alpha_b-1} , \quad \beta_g = \frac{(1-z)\tilde{\kappa}}{(1+c)^2\alpha_b} .
\end{aligned} \tag{2.92}$$

2.5.2 Final-state branching

Next consider emission from the outgoing parton c. In this case we write

$$q_i = \alpha_i p_c + \beta_i p_b + q_{\perp i} \tag{2.93}$$

To preserve $p_a = q_c + q_g - q_b$ we require

$$\alpha_c + \alpha_g - \alpha_b = \beta_b - \beta_c - \beta_g = 1 \tag{2.94}$$

whereas the mass-shell condition is now

$$\alpha_i\beta_i\,(Q^2+m_c^2) = \mathbf{q}_{\perp i}{}^2 + q_i^2 - \alpha_i^2 m_c^2 . \tag{2.95}$$

The new variables for emission from an outgoing parton are as in eqs. (2.38,2.39) with b replaced by c:

$$z = \frac{\alpha_c}{\alpha_c+\alpha_g} , \quad \tilde{\kappa} \equiv \frac{\tilde{q}^2}{Q^2} = \frac{1}{z^2}\left[c+\frac{\kappa}{(1-z)^2}\right] . \tag{2.96}$$

Thus in this case we find

$$\alpha_b = 0 , \quad \alpha_c = z , \quad \alpha_g = 1-z ,$$

$$\begin{aligned}\beta_b &= \frac{1}{1+c}[1+c+z(1-z)\tilde{\kappa}] ,\\ \beta_c &= \frac{1-z}{1+c}[2c+z(1-z)\tilde{\kappa}] ,\\ \beta_g &= \frac{1-z}{1+c}[z^2\tilde{\kappa}-2c] .\end{aligned} \tag{2.97}$$

2.5.3 Phase space variables

In this process the invariant phase space variables are usually taken to be

$$x_p = \frac{Q^2}{2p_a\cdot q_b} , \quad z_p = \frac{q_c\cdot q_b}{p_a\cdot q_b} . \tag{2.98}$$

In terms of the new variables for emission from the beam parton, we have

$$x_p = \frac{1}{(1+c)\alpha_b} = 2z\left(1+c+(1-z)\tilde{\kappa}+\sqrt{[1+c+(1-z)\tilde{\kappa}]^2-4z(1-z)\tilde{\kappa}}\right)^{-1} \tag{2.99}$$

$$z_p = (1+c)\beta_c = \frac{1}{2}\left(1-c-(1-z)\tilde{\kappa}+\sqrt{[1+c+(1-z)\tilde{\kappa}]^2-4z(1-z)\tilde{\kappa}}\right) \tag{2.100}$$

with the Jacobian

$$\frac{\partial(x_p,z_p)}{\partial(z,\tilde{\kappa})} = \frac{1}{\tilde{\kappa}}\left(\frac{1}{x_p}+\frac{1+c}{1-z_p}-2\right)^{-1} . \tag{2.101}$$

In the soft limit $z = 1-\epsilon$ we therefore find for the beam jet

$$x_p \sim \frac{1}{1+c}\left[1-\epsilon-\frac{\epsilon c\tilde{\kappa}}{(1+c)^2}\right] , \quad z_p \sim 1-\frac{\epsilon\tilde{\kappa}}{1+c} \tag{2.102}$$

and

$$\frac{\partial(x_p,z_p)}{\partial(z,\tilde{\kappa})} \sim \frac{\epsilon}{(1+c)^2} . \tag{2.103}$$

In terms of the variables for emission from the outgoing parton,

$$x_p = \frac{1}{(1+c)\beta_b} = \frac{1}{1+c+z(1-z)\tilde{\kappa}} , \quad z_p = \alpha_c = z , \tag{2.104}$$

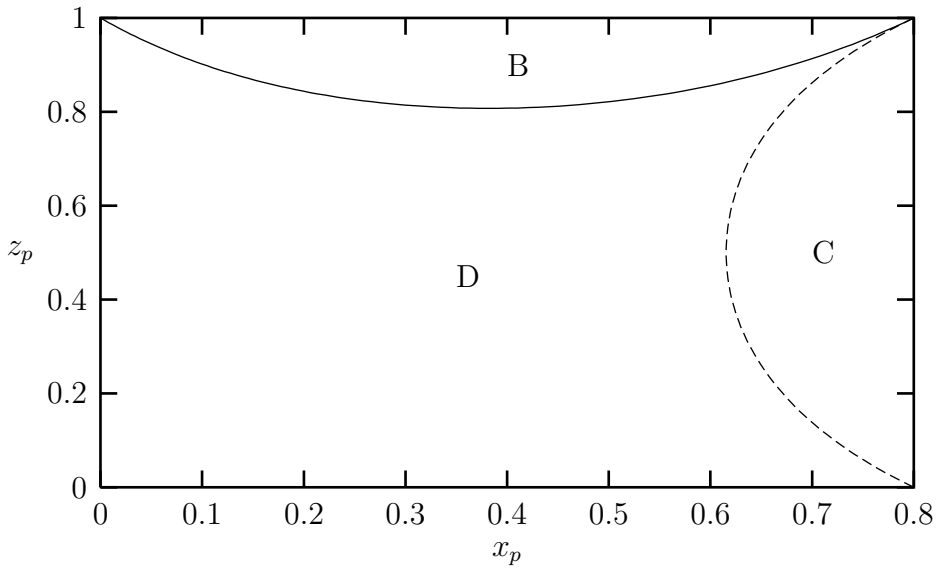


Figure 2.9: Beam jet (B), outgoing jet (C) and dead region (D) in initial-final state branching: $c = 0.25$, $\tilde{\kappa}_b = 1.25$, $\tilde{\kappa}_c = 1.5$.

so the Jacobian is simply

$$\frac{\partial(x_p, z_p)}{\partial(z, \tilde{\kappa})} = z(1-z)x_p^2 \,, \tag{2.105}$$

and in the soft limit

$$x_p \sim \frac{1}{1+c}\left[1 - \frac{\epsilon\tilde{\kappa}}{1+c}\right] \,, \quad z_p \sim 1 - \epsilon \,, \tag{2.106}$$

with the Jacobian again given by (2.103). For full coverage of phase space in the soft limit we require $\tilde{\kappa} < \tilde{\kappa}_b$ in jet b and $\tilde{\kappa} < \tilde{\kappa}_c$ in jet c, where

$$\tilde{\kappa}_b(\tilde{\kappa}_c - c) = (1+c)^2 \,. \tag{2.107}$$

Thus the most symmetrical choice is $\tilde{\kappa}_b = 1 + c$, $\tilde{\kappa}_c = 1 + 2c$, as shown in figure 2.9. On the other hand, any larger or smaller combination satisfying (2.107) is allowed, as illustrated in figure 2.10 for $\tilde{\kappa}_b = 10$.

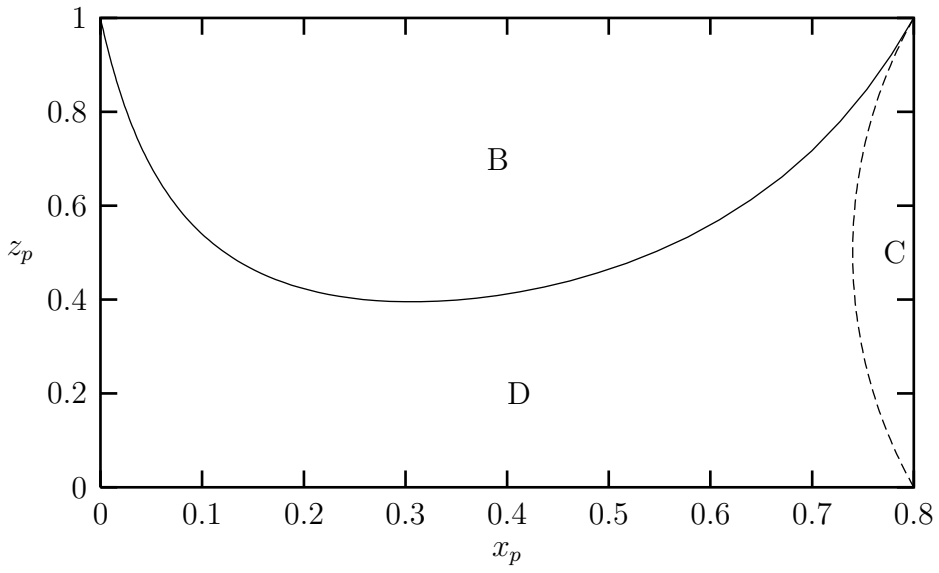


Figure 2.10: Beam jet (B), outgoing jet (C) and dead region (D) in initial-final state branching: $c = 0.25$, $\tilde{\kappa}_b = 10$, $\tilde{\kappa}_c = 0.40625$.

2.5.4 Example: deep inelastic scattering

Consider deep inelastic scattering on a hadron of momentum P^μ by exchange of a virtual photon of momentum q^μ. If the contribution to the Born cross section from scattering on a quark of momentum fraction $x_B = Q^2/2P\cdot q$ is represented by σ_0 (a function of x_B and Q^2), then the correction due to single gluon emission is given by

$$\frac{1}{\sigma_0}\frac{d^2\sigma}{dx_p\,dz_p} = \frac{C_F\alpha_{\rm S}}{2\pi}\frac{D(x_B/x_p)}{D(x_B)}\frac{1+(x_p+z_p-1)^2}{x_p(1-x_p)(1-z_p)}\,. \tag{2.108}$$

In the soft limit $x_p, z_p \to 1$ we have, from eqs. (2.102,2.106) with $c = 0$,

$$(1-x_p)(1-z_p) \sim \epsilon^2\tilde{\kappa} \tag{2.109}$$

and so

$$\frac{1}{\sigma_0}\frac{d^2\sigma}{dx_p\,dz_p} \sim \frac{C_F\alpha_{\rm S}}{\pi}\frac{1}{\epsilon^2\tilde{\kappa}}\,, \tag{2.110}$$

whereas the parton shower approximation gives

$$\frac{1}{\sigma_0}\frac{d^2\sigma}{dz\,d\tilde{\kappa}} \sim \frac{C_F\alpha_{\rm S}}{\pi}\frac{1}{\epsilon\tilde{\kappa}}\ . \tag{2.111}$$

Since the Jacobian factor (2.101) or (2.105) in this limit is simply ϵ, the shower approximation is exact in the soft limit.

2.5.5 Example: $q\bar{q} \to t\bar{t}$

We denote the momenta in this process by $p_a + p_b \to p_c + p_d$ and the $2 \to 2$ invariants by

$$\bar{s} = 2p_a \cdot p_b\ , \quad \bar{t} = -2p_a \cdot p_c\ , \quad \bar{u} = -2p_a \cdot p_d\ , \tag{2.112}$$

so that $\bar{s} + \bar{t} + \bar{u} = 0$. Colour flows from q to t and anticolour from $\bar{q}$ to $\bar{t}$. Therefore the momentum transfer $q = p_a - p_c = p_d - p_b$ is carried by a colour singlet and we preserve this 4-momentum during showering.

For emission from the incoming light quark or the outgoing top quark, we work in the Breit frame for this system, where

$$q = Q(0, \mathbf{0}, 1)\ , \quad p_a = \frac{1}{2}Q(1 + c, \mathbf{0}, 1 + c)\ , \quad p_c = \frac{1}{2}Q(1 + c, \mathbf{0}, -1 + c) \tag{2.113}$$

with $Q^2 = -\bar{t} - m_t^2$ and $c = m_t^2/Q^2$. Then the treatment of sects. 2.5.1 and 2.5.2 can be applied directly, with the substitution $b \to a$ since the emitting system is now (a, c) rather than (b, c). However, the phase space variables are no longer those of sect. 2.5.3 since they involve the momenta of the $\bar{q}$ and $\bar{t}$, which in the frame (2.113) take the general form

$$p_b \ = \ [\frac{1}{2}Q\sqrt{(1 + c)^2 + 4K}, \mathbf{Q}_\perp, -\frac{1}{2}Q(1 + c)]\ ,$$

$$p_d \;=\; [\frac{1}{2}Q\sqrt{(1+c)^2+4K}, \mathbf{Q}_\perp, \frac{1}{2}Q(1-c)] \;, \tag{2.114}$$

where $K = {\mathbf{Q}_\perp}^2/Q^2$ is related to the $2 \to 2$ invariants:

$$\bar{s} = \frac{1}{2}Q^2(1+c)^2\left[1+\sqrt{1+\frac{4K}{(1+c)^2}}\right] \;, \quad \bar{t} = -Q^2(1+c) \;, \quad \bar{u} = -\bar{s}-\bar{t} \;, \tag{2.115}$$

and so

$${\mathbf{Q}_\perp}^2 = -\bar{s}\left[1+\frac{\bar{s}}{(1+c)\bar{t}}\right] \;. \tag{2.116}$$

For emission from the incoming light quark we define as in sect. 2.5.1

$$q_i = \alpha_i p_a + \beta_i n + q_{\perp i} \tag{2.117}$$

for $i = a, c, g$, where $\mathbf{q}_{\perp a} = \mathbf{0}$, $\mathbf{q}_{\perp c} = -\mathbf{k}_\perp$, $\mathbf{q}_{\perp g} = \mathbf{k}_\perp$, and n is as in (2.88). Then the α_i's and β_i's are given by eqs. (2.92) with the substitution $b \to a$. The light antiquark and the antitop are not affected and therefore $q_b = p_b$, $q_d = p_d$. This allows the complete kinematics of the $2 \to 3$ process to be reconstructed. The $2 \to 3$ invariants can be defined as in ref. [44]:

$$s = 2q_a\cdot q_b \;, \quad t_1 = -2q_a\cdot q_c \;, \quad t_2 = -2q_b\cdot q_d \;, \quad u_1 = -2q_a\cdot q_d \;, \quad u_2 = -2q_b\cdot q_c \;. \tag{2.118}$$

It is convenient to express $n = p_c - cq$ so that (for $i = a, c, g$)

$$q_i = (\alpha_i - c\beta_i)p_a + (1+c)\beta_i p_c + q_{\perp i} \;. \tag{2.119}$$

Then we find

$$s = \alpha_a\bar{s} \;, \quad t_1 = \alpha_a\beta_c(1+c)\bar{t} \;, \quad t_2 = \bar{t} \;, \quad u_1 = \alpha_a\bar{u} \;, \quad u_2 = \beta_c(\bar{u}-c\bar{t}) - \alpha_c\bar{s} - 2\mathbf{k}_\perp\cdot\mathbf{Q}_\perp \;. \tag{2.120}$$

For emission from the outgoing top we use the results of sect. 2.5.2, again with the substitution $b \to a$. Thus we now have for $i = a, c, g$

$$q_i = \alpha_i p_c + \beta_i p_a + q_{\perp i} \tag{2.121}$$

where the α_i's and β_i's are given by eqs. (2.97) with $b \to a$, and we find that

$$s = \beta_a \bar{s} \ , \quad t_1 = \alpha_c \beta_a \bar{t} \ , \quad t_2 = \bar{t} \ , \quad u_1 = \beta_a \bar{u} \ , \quad u_2 = \alpha_c \bar{u} - \beta_c \bar{s} - 2\mathbf{k}_\perp \cdot \mathbf{Q}_\perp \ . \tag{2.122}$$

Similar formulae to eqs. (2.120) and (2.122), with the replacements $a \to b$ and $c \to d$, will hold for the case of gluon emission from the colour-connected $(\bar{q}\bar{t})$ system. Using these relations, one can study the distribution of gluon radiation in the parton shower approximation and compare it with the exact $q\bar{q} \to t\bar{t}g$ matrix element. Agreement will be good in the soft and/or collinear regions but there will be regions of hard, wide-angle gluon emission in which matrix element corrections should be applied. Alternatively, the above equations can be used to formulate a modified subtraction scheme for combining fixed-order and parton-shower results, as was done in ref. [44] for a different parton-shower algorithm.

2.6 Decay colour connection

Consider the process $b \to ca$ where a is a colour singlet and the decaying parton b and outgoing parton c are colour-connected. Examples are bottom quark decay, $b \to cW^*$, and top decay, $t \to bW$. Here we have to preserve the 4-momentum of the decaying parton b and therefore we work in its rest frame,

$$p_b = m_b(1, \mathbf{0}, 0) \ , \quad p_c = \frac{1}{2} m_b(1 - a + c, \mathbf{0}, \lambda) \ , \quad p_a = \frac{1}{2} m_b(1 + a - c, \mathbf{0}, -\lambda) \ , \tag{2.123}$$

where $a = m_a^2/m_b^2$, $c = m_c^2/m_b^2$ and now

$$\lambda = \lambda(1, a, c) = \sqrt{(1 + a - c)^2 - 4a} = \sqrt{(1 - a + c)^2 - 4c} \, . \tag{2.124}$$

2.6.1 Initial-state branching

For emission of a gluon g from the decaying parton b we write

$$q_i = \alpha_i p_b + \beta_i n + q_{\perp i} \tag{2.125}$$

where $\mathbf{q}_{\perp g} = \mathbf{k}_\perp$, $\mathbf{q}_{\perp c} = -\mathbf{k}_\perp$, $\mathbf{q}_{\perp b} = \mathbf{0}$ and we choose

$$n = \frac{1}{2} m_b (1, \mathbf{0}, 1) \, , \tag{2.126}$$

i.e. aligned along p_c in the rest frame of b. The mass-shell conditions give

$$\beta_a = \frac{a}{\alpha_a} - \alpha_a \, , \quad \beta_c = \frac{c + \kappa}{\alpha_c} - \alpha_c \, , \quad \beta_g = \frac{\kappa}{\alpha_g} - \alpha_g \, , \tag{2.127}$$

with $\kappa = {\mathbf{k}_\perp}^2/m_b^2$. From momentum conservation

$$\alpha_a + \alpha_c + \alpha_g = \frac{a}{\alpha_a} + \frac{c + \kappa}{\alpha_c} + \frac{\kappa}{\alpha_g} = 1 \, . \tag{2.128}$$

Recall that in initial-state branching of a heavy object our new evolution variable is given by (2.20), so we have

$$\alpha_g = 1 - z \, , \quad \kappa = (\tilde{\kappa} - 1)(1 - z)^2 \tag{2.129}$$

where $\tilde{\kappa} = \tilde{q}^2/m_b^2 > 1$. Introducing for brevity the notation

$$w = 1 - (1-z)(\tilde{\kappa}-1) \;, \quad u = 1 + a - c - (1-z)\tilde{\kappa} \;, \quad v = \sqrt{u^2 - 4awz} \;, \tag{2.130}$$

from (2.128) we find

$$\alpha_a = \frac{u+v}{2w} \;, \quad \alpha_c = 1 - \alpha_a - \alpha_g = z - \alpha_a \;. \tag{2.131}$$

2.6.2 Final-state branching

For radiation from the outgoing parton c we write

$$q_i = \alpha_i p_c + \beta_i n + q_{\perp i} \tag{2.132}$$

where p_c is given by (2.123). Since the colour-connected parton b is at rest in our working frame of reference, the choice of the light-like vector n in this case is somewhat arbitrary. By analogy with the cases treated earlier, we choose it to be opposite to that used for the radiation from b, i.e. along the direction of the colour singlet a:

$$n = \frac{1}{2} m_b (\lambda, \mathbf{0}, -\lambda) \;. \tag{2.133}$$

The kinematics are then identical with those for final-final connection (sect. 2.3), with the replacement $b \to c$, $c \to a$.

2.6.3 Phase space variables

As in sect. 2.3, it is convenient to use the Dalitz plot variables, which in this case are

$$x_i = \frac{2q_i \cdot p_b}{m_b^2} \ . \tag{2.134}$$

For emission from the decaying parton b we have $x_i = 2\alpha_i + \beta_i$ and hence, from (2.131),

$$x_a = \frac{u+v}{2w} + \frac{u-v}{2z} \ , \quad x_c = w + z - x_a \ , \quad x_g = 2 - w - z = (1-z)\tilde{\kappa} \ , \tag{2.135}$$

with the Jacobian factor

$$\frac{\partial(x_a, x_g)}{\partial(z, \tilde{\kappa})} = (1-z)\left[\frac{u+v}{2w^2} - \frac{u-v}{2z^2} + \frac{a(w-z)^2}{vwz}\right] \ . \tag{2.136}$$

In the soft limit $z \to 1-\epsilon$ we find

$$x_a \sim 1 + a - c - \epsilon\tilde{\kappa}_b' \ , \quad x_g \sim \epsilon\tilde{\kappa}_b \tag{2.137}$$

where

$$\tilde{\kappa}_b' = \lambda + \frac{\tilde{\kappa}_b}{2}(1 - a + c - \lambda) \ . \tag{2.138}$$

For emission from the outgoing parton c we have, from (2.42) with the replacement $b \to c$, $c \to a$:

$$\begin{aligned} x_a &= 1 + a - c - z(1-z)\tilde{\kappa} \\ x_c &= (2 - x_a)r + (z - r)\sqrt{x_a^2 - 4a} \\ x_g &= (2 - x_a)(1 - r) - (z - r)\sqrt{x_a^2 - 4a} \end{aligned} \tag{2.139}$$

where

$$r = \frac{1}{2}\left(1 + \frac{c}{1+a-x_a}\right) . \tag{2.140}$$

In the soft limit we have from (2.50)

$$x_a \sim 1 + a - c - \epsilon\tilde{\kappa}_c , \quad x_g \sim \epsilon\tilde{\kappa}'_c \tag{2.141}$$

where

$$\tilde{\kappa}'_c = \lambda + \frac{\tilde{\kappa}_c}{2c}(1 - a + c - \lambda) . \tag{2.142}$$

For full coverage of the soft region we require

$$\frac{\tilde{\kappa}_b}{\tilde{\kappa}'_b} = \frac{\tilde{\kappa}'_c}{\tilde{\kappa}_c} \tag{2.143}$$

which gives in this case

$$(\tilde{\kappa}_b - 1)(\tilde{\kappa}_c - c) = \frac{1}{4}(1 - a + c + \lambda)^2 . \tag{2.144}$$

Note that, while there is no upper limit on $\tilde{\kappa}_b$, the largest value that can be chosen for $\tilde{\kappa}_c$ is given by the equivalent of (2.55),

$$\tilde{\kappa}_c < 4(1 + a - 2\sqrt{c} - c) . \tag{2.145}$$

2.6.4 Example: top decay

In the decay $t \to Wbg$ we have $a = m_W^2/m_t^2 = 0.213$ and $c = m_b^2/m_t^2 = 0.026$, so for simplicity we neglect c. Then for radiation from the top we have from (2.135)

$$x_W = \frac{u+v}{2w} + \frac{u-v}{2z} , \quad x_g = (1-z)\tilde{\kappa} , \tag{2.146}$$

where u, v, w are given by eqs. (2.130) with $c \to 0$. The phase space is the region

$$0 < x_g < 1 - a \ , \quad 1 - x_g + \frac{a}{1 - x_g} < x_W < 1 + a \ . \tag{2.147}$$

Notice that for real x_W we require $u^2 > 4awz$, i.e.

$$1 < \tilde{\kappa} < 1 + a \left[1 - \sqrt{\frac{z(1-a)}{a(1-z)}} \right]^2 \ , \tag{2.148}$$

and

$$1 - \frac{1-a}{\tilde{\kappa} + 2\sqrt{a(\tilde{\kappa} - 1)}} < z < 1 \ . \tag{2.149}$$

Thus there is no upper limit on $\tilde{\kappa}$, but the range of z becomes more limited as $\tilde{\kappa}$ increases.

For radiation from the b we have from (2.139)

$$\begin{aligned} x_W &= 1 + a - z(1-z)\tilde{\kappa} \\ x_g &= \frac{1}{2}(2 - x_W) - (z - \frac{1}{2})\sqrt{x_W^2 - 4a} \ . \end{aligned} \tag{2.150}$$

To cover the soft region we require $\tilde{\kappa} < \tilde{\kappa}_t$ for emission from the top quark and $\tilde{\kappa} < \tilde{\kappa}_b$ for that from the bottom, where (2.144) gives

$$\tilde{\kappa}_b = \frac{(1-a)^2}{\tilde{\kappa}_t - 1} \ . \tag{2.151}$$

The most symmetrical choice would therefore appear to be $\tilde{\kappa}_b = \tilde{\kappa}_t - 1 = 1 - a = 0.787$, as illustrated in figure 2.11.

As mentioned above, there is no upper limit on $\tilde{\kappa}_t$. Thus the region covered by gluon emission from the top quark can be as large as we like. However, (2.145) tells us that

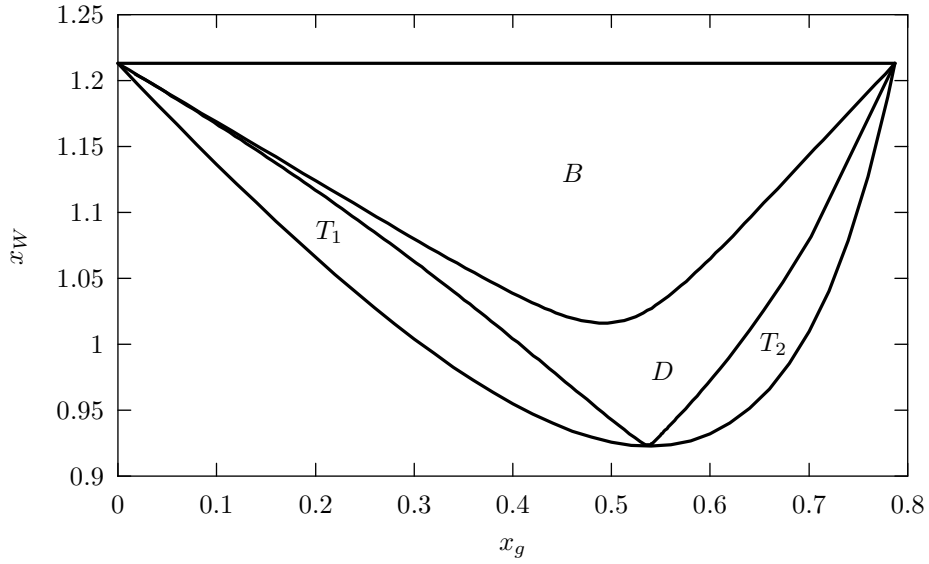


Figure 2.11: Phase space for decay $t \to Wbg$, with symmetric choice of emission regions for the b (B) and the t (T_1,T_2), and the dead region (D).

the upper limit for radiation from the b is

$$\tilde{\kappa}_b < 4(1-\sqrt{a})^2 = 1.16 \,, \tag{2.152}$$

and correspondingly $\tilde{\kappa}_t > 1 + \frac{1}{4}(1+\sqrt{a})^2 = 1.53$. Figure 2.12 shows this maximal region that can be covered by emission from the b, together with the complementary regions of emission from the t.

We note from figs. 2.11 and 2.12 that, for any value of $\tilde{\kappa}_t$, the region for emission from the top quark consists of two distinct parts that touch at the point $x_g = 1-\sqrt{a}$, $x_W = 2\sqrt{a}$, where the W boson is at rest: a subregion T_1 which includes the soft limit $x_g \to 0$ and a hard gluon region T_2.

The exact $t \to Wbg$ differential decay rate to first order in α_s is given in [56]:

$$\frac{1}{\Gamma_0}\frac{d^2\Gamma}{dx_W dx_g} = \frac{\alpha_s}{\pi}\frac{C_F}{(1+a-x_W)x_g^2}\left\{ x_g - \frac{(1+a-x_W)(1-x_g)+x_g^2}{1-a}\right.$$

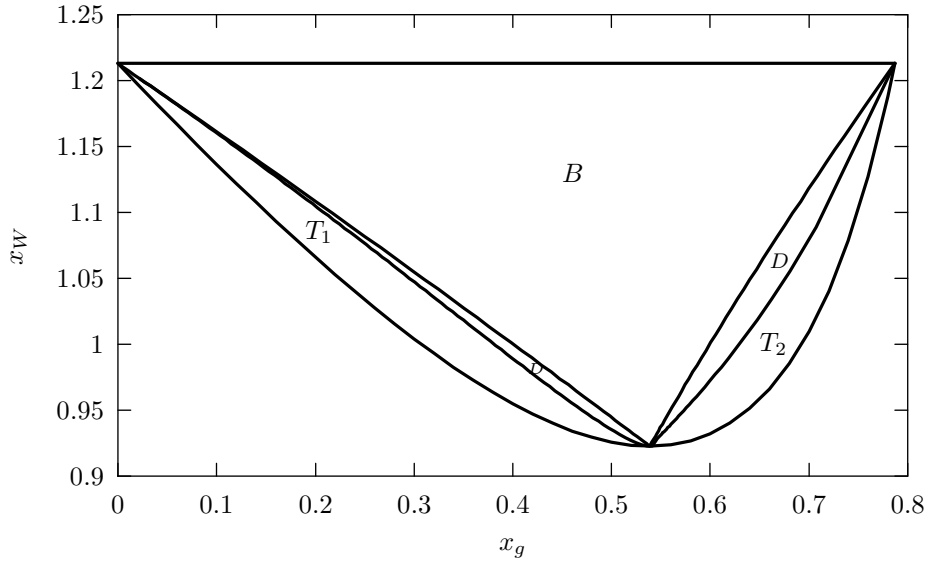


Figure 2.12: Phase space for decay $t \to Wbg$, with maximal region (B) for emission from the b, together with complementary regions of emission from the t (T_1,T_2) and the dead region (D).

$$+x_g \frac{(x_W + x_g - 1 - a)^2}{2(1-a)^2} + \frac{2a(1+a-x_W)x_g^2}{(1-a)^2(1+2a)} \Bigg\} \tag{2.153}$$

where Γ_0 is the lowest-order decay rate. In the soft region $x_W \to 1+a$, $x_g \to 0$ this becomes

$$\frac{1}{\Gamma_0}\frac{d^2\Gamma}{dx_W dx_g} \sim \frac{\alpha_{\rm S}}{\pi}\frac{C_F}{x_g}\left[\frac{1}{1+a-x_W} - \frac{1}{(1-a)x_g}\right] . \tag{2.154}$$

For soft gluon emission ($1-z = \epsilon \to 0$) from the top quark we have from eqs. (2.137,2.138)

$$x_W \sim 1 + a - \epsilon(1-a) , \quad x_g \sim \epsilon\tilde{\kappa} \tag{2.155}$$

and so the exact form of the soft gluon distribution is

$$\frac{1}{\Gamma_0}\frac{d^2\Gamma}{dx_g dx_W} \sim \frac{\alpha_s}{\pi}\frac{C_F}{\epsilon^2} f_t(\tilde{\kappa}) \tag{2.156}$$

where

$$f_t(\tilde{\kappa}) = \frac{\tilde{\kappa} - 1}{(1-a)\tilde{\kappa}^2} \ . \tag{2.157}$$

In the same region the parton shower approximation (2.12) gives

$$\frac{d^2P}{dx_g dx_W} \sim \frac{1}{(1-a)\epsilon} \frac{d^2P}{dz\, d\tilde{\kappa}} \sim \frac{\alpha_s}{\pi} \frac{C_F}{(1-a)\epsilon^2\tilde{\kappa}} \left(1 - \frac{1}{\tilde{\kappa}}\right) = \frac{\alpha_s}{\pi} \frac{C_F}{\epsilon^2} f(\tilde{\kappa}) \ . \tag{2.158}$$

Thus we see that, for emission from the top quark, the shower approximation is exact in the soft limit. At higher gluon energies, inside the region T_1 the parton shower overestimates the exact matrix element and can therefore be corrected easily by the rejection method. In the hard gluon region T_2, which contributes only a small finite correction to the cross section, the parton shower overestimates the matrix element at lower values of x_g but underestimates it at the highest values. Therefore a combination of rejection and matrix element correction is needed in this region.

For emission from the bottom quark in the soft limit, we use the results of sect. 2.3.2 with the substitution $b \to 0$, $c \to a$ to obtain

$$x_W \sim 1 + a - \epsilon\tilde{\kappa} \ , \quad x_g \sim \epsilon \left(1 - a + \frac{\tilde{\kappa}}{1-a}\right) \ . \tag{2.159}$$

Therefore the exact soft gluon distribution in the b jet should be

$$\frac{1}{\Gamma_0} \frac{d^2\Gamma}{dx_g dx_W} \sim \frac{\alpha_s}{\pi} \frac{C_F}{\epsilon^2} f_b(\tilde{\kappa}) \tag{2.160}$$

where

$$f_b(\tilde{\kappa}) = \frac{1}{(1-a)\tilde{\kappa}} \left[1 + \frac{\tilde{\kappa}}{(1-a)^2}\right]^{-2} \ . \tag{2.161}$$

On the other hand the parton shower approximation in this case gives simply

$$\frac{d^2P}{dx_g dx_W} \sim \frac{1}{(1-a)\epsilon} \frac{d^2P}{dz\, d\tilde{\kappa}} \sim \frac{\alpha_s}{\pi} \frac{C_F}{(1-a)\epsilon^2\tilde{\kappa}} \ . \tag{2.162}$$

Thus the soft gluon distribution in the b jet region is overestimated by a factor of

$$\left[1 + \frac{\tilde{\kappa}}{(1-a)^2}\right]^2 , \tag{2.163}$$

which can be corrected by the rejection method. This factor varies from 1 to 5.2 for the symmetric choice of the b jet region $\tilde{\kappa}_b = 1 - a$ depicted in figure 2.11. For the maximal b jet shown in figure 2.12, it rises to 8.3. Since the shower approximation is exact in the soft limit for emission from the top, one can reduce the amount of soft correction required by decreasing the b jet region and increasing that for top emission, in accordance with (2.151). However, for large values of $\tilde{\kappa}_t$ the dead region moves near to the collinear singularity at $x_W = 1 + a$ and a large hard matrix element correction becomes necessary.

2.7 Conclusions

A new formulation of the parton-shower approximation to QCD matrix elements has been presented. This formalism offers a number of advantages over previous ones. Direct angular ordering of the shower ensures a good emulation of important QCD coherence effects, while the connection between the shower variables and the Sudakov-like representation of momenta (2.1) simplifies the kinematics and their relation to phase space invariants. The use of mass-dependent splitting functions with the new variables allows an accurate description of soft gluon emission from heavy quarks over a wide angular region, including the collinear direction. The separation of showering into contributions from pairs of colour-connected hard partons permits a general treatment of coherence effects, which should be reliable at least to leading order in the number of colours. Since the formulation is slightly different for initial- and final-state showering, formulae for all colour-connected combinations of incoming and outgoing partons have been given.

This new shower formulation is a key element of the event generator Herwig++ [41]. Chapter 4 and 5 describe how this new formalism is implemented and complete results for e^+e^- annihilation are presented.

Chapter 3

Hadronization

Hadronization is the process in which the perturbative partons (quarks and gluons) from the shower enter the non-perturbative phase and are converted into the observed hadrons. This is not well understood and instead is modelled by a hadronization model. As discussed earlier there are a few different types of hadronization models. Pythia [25] uses a string fragmentation model whereas HERWIG 6.5 [24] uses a cluster hadronization model. Here I describe this cluster model and the modifications that have been made in the new Herwig++ hadronization model.

3.1 Cluster Formation

The shower terminates and leaves colour connected pairs with low virtuality. These partons need to combine to form hadrons. The first step of the cluster hadronization is to form clusters out of these colour connected particles. These clusters are made up of quark–anti-quark pairs or (anti-)diquark– (anti-)quark pairs. This section presents each step of the process of cluster formation up until the decay of the clusters into hadrons.

3.1.1 Gluon Splitting

The initial step of the cluster hadronization is to split the gluons into quark–anti-quark pairs. Since at the end of the shower the gluons are put on-shell, they must be given a mass, m_g, in order to decay into a quark–anti-quark pair. The default value of this mass is 0.750 GeV. This is then high enough to isotropically decay into $u\bar{u}$ and $d\bar{d}$ pairs. Each gluon is randomly assigned a decay into either u quarks or d quarks and is decayed uniformly in $\cos\theta$ and ϕ. After the isotropic decays the event is left with only colour connected (di)quarks and anti-(di)quarks.

3.1.2 Cluster Formation

Clusters are themselves colour-singlets. They can be made up of either two partons (quark–anti-quark pair) or of three partons (quarks or anti-quarks all of different colours). In the cluster model of Herwig++ a three parton cluster will occur in baryon non-conserving events, which have colour sources or sinks, and in events where a beam particle is a baryon. This can be seen because a cluster composed of three partons has baryon number ± 1. Since drawing from the vacuum doesn't change the baryon number, the decay of these clusters must have baryon number ± 1. Unless the cluster is derived from a beam baryon, these types of clusters must only occur in baryon non-conserving events.

A cluster is formed simply by finding a colour-singlet pair (or triplet) of partons. The 4-momentum of the cluster is just the sum of the momenta of the partons. Clusters are created for all sets of colour singlets. If a quark or anti-quark is created from a colour source or sink (baryon-violating) it forms a three parton cluster with its colour neighbours. Two of these partons are randomly combined into a diquark for the purpose of the cluster decay.

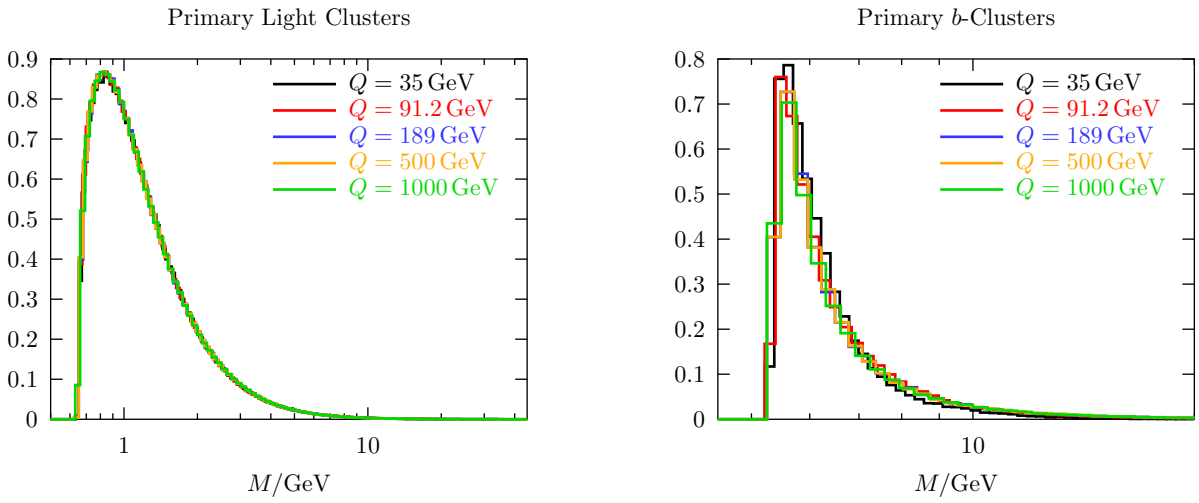


Figure 3.1: Primary cluster mass distribution in the e^+e^- annihilation at various centre-of-mass energies, Q, for clusters containing only light quarks (left) and a b quark (right).

The principle of colour-preconfinement says that the invariant mass distribution of the clusters is independent of the centre-of-mass energy. This idea is needed to fully separate the perturbative regime from the non-perturbative regime. Figure 3.1 shows this distribution for Herwig++. The first plot in figure 3.1 shows the distribution for the light clusters. The second plot in figure 3.1 shows the distribution for the b quarks only. This shows that the fall off of the distributions is similar for each flavour.

3.1.3 Cluster Fission

The mass of a cluster is given by $M^2 = p^2$ of the cluster. In order for the shower to combine more smoothly with the hadronization, clusters of a large mass are decayed into two new clusters. If the shower were to be cut off at a larger scale then there would be fewer more energetic partons. On the other hand if it were to be cut off at a smaller scale it would produce many lower energy partons. Splitting the clusters with large mass into two clusters with smaller mass allows the hadron multiplicity to be much less variable with the shower cutoff. In turn this allows the tuning of the shower to not be as dependant on the hadron multiplicity, making for more consistent results.

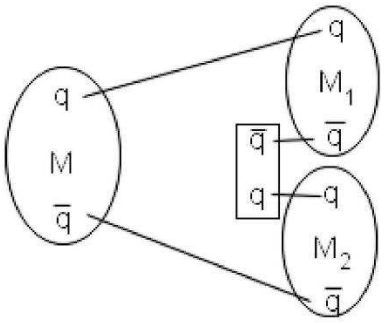


Figure 3.2: Cluster Fission. Cluster of mass M decays into two new clusters of masses M_1 and M_2 by drawing a pair from the vacuum.

A cluster is split into two clusters if the mass does not satisfy the condition

$$M^{\texttt{Cl}_{\texttt{pow}}} < \texttt{Cl}_{\texttt{max}}{}^{\texttt{Cl}_{\texttt{pow}}} + \Sigma_c^{\texttt{Cl}_{\texttt{pow}}}, \tag{3.1}$$

where $\texttt{Cl}_{\texttt{pow}}$ and $\texttt{Cl}_{\texttt{max}}$ are parameters and Σ_c is the sum of the masses of the constituents that make up the cluster.

If a cluster is to be split a $q\bar{q}$ pair is chosen from the vacuum. Only u, d or s flavours are chosen with probabilities given by the parameters $\texttt{Pwt}_i$ with i being the flavour. Once a pair is chosen from the vacuum the cluster is decayed into two new clusters with one of the original partons in each cluster. A schematic of this decay is shown in figure 3.2. In the case where the partons are not beam remnants, the mass distribution of the new clusters are given by

$$M_1 = m_1 + (M - m_1 - m_3)\, r_1^{1/P}, \tag{3.2}$$

$$M_2 = m_2 + (M - m_2 - m_3)\, r_2^{1/P} \tag{3.3}$$

where m_1 and m_2 are the two components of the original cluster, m_3 is the mass of the parton drawn from the vacuum, M_1, M_2 are the new cluster masses and r_i are random numbers in the interval $[0, 1]$. P is another parameter of the model. If the parton is not a b quark then P is `PSplt1` otherwise it is `PSplt2` (see table 4.1). Points are only chosen

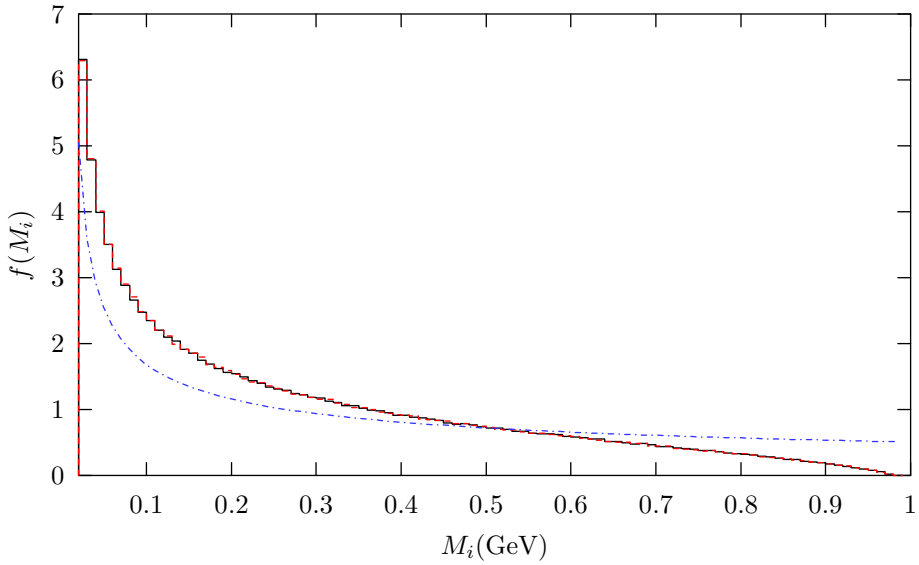


Figure 3.3: The distribution of given in (3.5) (blue, dash-dotted) and the actual correlated distributions (black/red, solid) when (3.4) are applied. This is for a cluster mass of 1 GeV, $P = 0.5$ and all three constituent masses $m_i = 0.01$ GeV.

in the range confined by

$$M > M_1 + M_2; \quad M_1 > m_1 + m_3; \quad M_2 > m_2 + m_3. \tag{3.4}$$

The corresponding probability density is

$$f(M_i) = N \frac{P(M_i - m_i)^{P-1}}{(M - m_i - m_3)^P}, \tag{3.5}$$

where N is an arbitrary normalization term. It is important to note that because of the constraint in (3.4) the distributions of the two clusters are correlated and therefore do not exactly follow (3.5). An example of this is shown in fig. 3.3

If parton i is from the beam remnant and there is no underlying event model being used, the cluster fissions with a different distribution. This distribution has one parameter $\texttt{BCl}_{\texttt{pow}}$, which is set to 1.0 GeV by default. This is used to define $b = 2/\texttt{BCl}_{\texttt{pow}}$ which

gives

$$r_{min} = e^{-bX}, \tag{3.6}$$

where $X = M - m_1 - m_2 - 2m_3$. The mass distribution is then defined by

$$f(M_i) = m_i + m_3 - \frac{\log R}{b}, \tag{3.7}$$

where

$$R = (r_{min} + r_1 (1 - 2r_{min})) (r_{min} + r_2 (1 - 2r_{min})) . \tag{3.8}$$

r_1 and r_2 are two flatly distributed random numbers. This mass distribution is designed to decrease rapidly to avoid splitting the cluster many times which would produce large transverse energies. This would not be desired as the beam remnant process is meant to be a soft process.

3.2 Cluster Decays

The last stage of the cluster hadronization model is the cluster decays. Here the clusters of a given flavour (q_1, q_2) draw a (di)quark anti-(di)quark pair $(q, \bar{q})$ from the vacuum and form a pair of hadrons with the combination $(q_1, \bar{q})$ and (q, q_2). The possible hadrons are selected based on spin, flavour and phase space. The exact way of accepting and rejecting these combinations are described in three variants: HERWIG 6.5 [24] , Kupco's method [57] and Herwig++. The method of Herwig++ is described in the next section.

3.2.1 HERWIG 6.5

The method of cluster decays implemented in HERWIG is described here. Before the software runs, it initialized a list of data for each type of hadron. This list contains two

weights,

- w: a spin weight for the hadron. This is $\frac{2s+1}{M_{q,q'}}$ where $M_{q,q'}$ is $2s+1$ for the largest spin of a flavour group. There are three of these groups which contain all the hadrons of a given flavor combination: (1) $u\bar{u}$ and $d\bar{d}$, (2) $s\bar{s}$ and (3) all the remaining flavours.

- s: this is a phase space correction weight which is normally set to unity.

HERWIG 6.5 chooses a mode in several phases. First a flavour is drawn out of the vacuum based on the probability

$$P_q = \frac{\texttt{Pwt}(q)}{\sum_q \texttt{Pwt}(q)}. \tag{3.9}$$

The values of $\texttt{Pwt}(q)$ are just parameters for u, d, s, c and b. For the diquarks, however, the weight is

$$P_{q_i q_j} = \frac{1}{2}\left(1+\delta_{ij}\right)\texttt{Pwt}_{qq}\texttt{Pwt}_{q_i}\texttt{Pwt}_{q_j}, \tag{3.10}$$

where $\texttt{Pwt}_{qq}$ is the diquark parameter.

The flavour combinations, (q_1, q) and (q, q_2), have the possibility of forming $N_{q_1,q}$ and N_{q,q_2} different hadrons, respectively. Hadrons $a_{q_1,q}$ and b_{q,q_2} are chosen randomly from this list. These hadrons have spin weight w_a and w_b and are rejected based on these weights.

Lastly, the resulting pair (a, b) are accepted or rejected based on phase space. The probability is given as

$$P_{phase}(a,b) = \frac{p^*_{a,b} s_a s_b}{p^*_{max}}, \tag{3.11}$$

where s_a and s_b are phase space adjusting parameters and p^*_{max} is the p^* value for the lightest hadron pair of the flavours $(q_1, q), (q, q_2)$. p^* for a two-body decay is the

c.m. momentum in the decay $M \to m_1 m_2$ and is given by

$$p^* = \frac{\sqrt{M^2 - (m_1 + m_2)^2}\sqrt{M^2 - (m_1 - m_2)^2}}{2M}, \tag{3.12}$$

in the region of valid phase space, $M \geq m_1 + m_2$. It is set to zero otherwise.

In the end all of this gives (approximately) the probability of choosing hadrons $a_{q_1,q}$ and b_{q,q_2} as

$$P(a_{q_1,q}, b_{q,q_2}|q_1, q_2) = P_q \frac{1}{N_{q_1,q}} \frac{1}{N_{q,q_2}} \frac{w_a}{M_{q_1,q}} \frac{w_b}{M_{q,q_2}} \frac{p^*_{a,b} s_a s_b}{p^*_{max}}. \tag{3.13}$$

Technically this isn't the probability because of the rejection scheme used in the algorithm but it is close and is able to illustrate the main advantages and disadvantages of the algorithm.

The problem with this method, as described by Kupco [57], is that as more hadrons are added to the list then the particular flavour content of the new hadrons is suppressed by the growing factor N. In effect, the probability of choosing a hadron of a given flavour is proportional to the average of all the p^*'s of the flavour. In most cases a majority of hadrons are inaccessible due to mass contraints. This leads to a suppression of the lighter hadrons of that flavour, even for clusters which are too light to decay into the new heavier states.

To look at this further we analyze the result of isospin symmetric clusters $u\bar{u}, d\bar{d}, u\bar{d}$ and $d\bar{u}$. For a cluster with mass just above threshold for the production of $\pi^0\pi^0$ and $\pi^+\pi^-$ these should be produced with the ratio $\pi^0 : \pi^+ : \pi^- = 1 : 1 : 1$. It is easy to see that the ratio between the states generated by $u\bar{d}$ and $d\bar{u}$ are $1 : 1$. Instead we look here at just the part of the ratio which differs from unity, $u\bar{u}$ and $d\bar{d}$. Using (3.13) we find (after assuming $s_a = s_b = 1$ and $p^*_{\pi^0,\pi^0} = p^*_{\pi^+,\pi^-} = p^*_{max}$)

$$P(\pi^0, \pi^0|u, \bar{u}) \;\; = \;\; P_u \frac{1}{N^2_{u\bar{u}}} \frac{w^2_{\pi^0}}{M^2_{u\bar{u}}}, \tag{3.14}$$

$$P(\pi^+,\pi^-|u,\bar{u}) = P_d\frac{1}{N_{d\bar{u}}}\frac{1}{N_{u\bar{d}}}\frac{w_{\pi^+}}{M_{u\bar{d}}}\frac{w_{\pi^-}}{M_{d\bar{u}}}, \quad (3.15)$$

$$P(\pi^0,\pi^0|d,\bar{d}) = P_d\frac{1}{N^2_{d\bar{d}}}\frac{w^2_{\pi^0}}{M^2_{d\bar{d}}}, \quad (3.16)$$

$$P(\pi^+,\pi^-|d,\bar{d}) = P_u\frac{1}{N_{d\bar{u}}}\frac{1}{N_{u\bar{d}}}\frac{w_{\pi^+}}{M_{u\bar{d}}}\frac{w_{\pi^-}}{M_{d\bar{u}}}. \quad (3.17)$$

The M's for $u\bar{u}$ and $d\bar{d}$ are equal and the M's for $u\bar{d}$ and $d\bar{u}$ are equal. We can also set the flavour probabilities to unity. Lastly, the spin weights of the π^+ and π^- are equal. We now find the ratio of pions as

$$\pi^0 : \pi^+ : \pi^- = \frac{2}{N^2_{u\bar{u}}}\frac{w^2_{\pi^0}}{M_{uu}} : \frac{1}{N^2_{u\bar{d}}}\frac{w^2_{\pi^+}}{M_{du}} : \frac{1}{N^2_{d\bar{u}}}\frac{w^2_{\pi^-}}{M_{ud}}. \quad (3.18)$$

Obviously, these ratios are dominated by the number of hadrons of a particular flavour content. It just happens that in HERWIG 6.5 there was the right number of $u\bar{u}, d\bar{d}$ and $u\bar{d}, d\bar{u}$ hadrons in the list to give approximately the correct ratios. Figure 3.4 shows the ratio for $\pi^0 : \pi^+$ as the cluster mass increases for $u\bar{u}$ and $d\bar{d}$ clusters. It can be seen that the ratios are dictated by the number of $u\bar{u}, d\bar{d}$ hadrons that are in the list.

3.2.2 Kupco Method

Kupco [57] was the first person to point out the problem of suppression of hadrons when new modes of the same flavours were added. He realized that the problem was that the probabilities were proportional to an average of the p^*'s of a particular flavour content.

In order to remedy the problem a new set of probabilities for choosing a decay mode was used. Instead of splitting the probability into independant parts, as was done in the original version, one weight was created for each hadronic mode.

$$W(a_{q_1,q}, b_{q,q_2}|q_1,q_2) = P_q w_a w_b s_a s_b p^*_{a,b}. \quad (3.19)$$

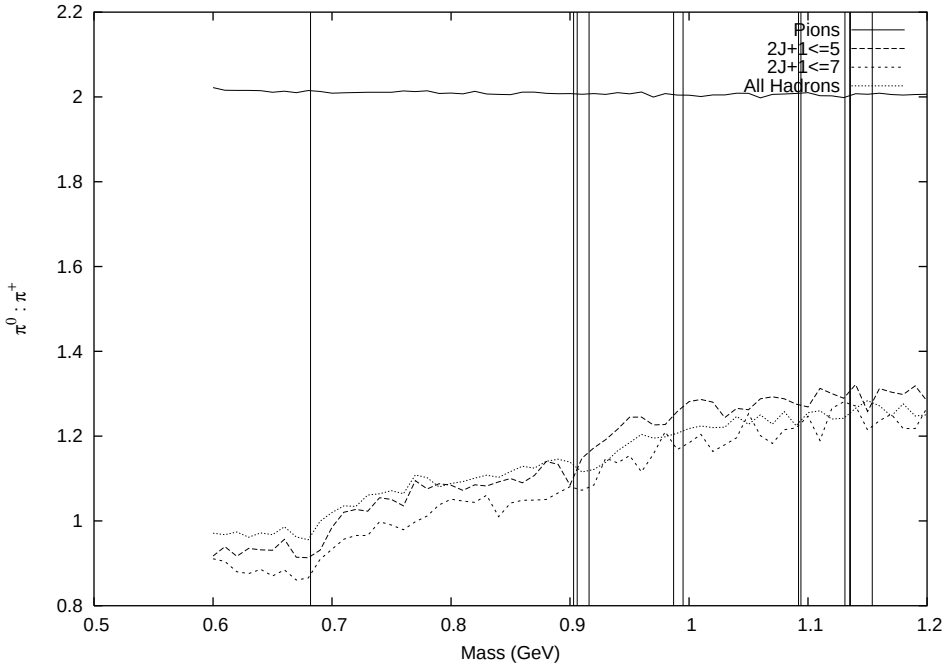


Figure 3.4: The ratio of π^0 to π^+ as the cluster mass is increased and with different number of hadrons in the $u\bar{u}$ list. The vertical bars indicate the threshold for a new set of hadrons to be produced.

The probability is then

$$P(a_{q_1,q}, b_{q,q_2}|q_1, q_2) = \frac{W(a_{q_1,q}, b_{q_2,q}|q_1, q_2)}{\sum_{c,d,q'} W(c_{q_1,q'}, d_{q_2,q'}|q_1, q_2)}. \tag{3.20}$$

The addition of new hadrons to the list now increases the probability of choosing that particular flavour. Because a majority of the hadrons are quite heavy the p^* for any mode with the new hadron is zero in many cases. Therefore, this new hadron has no effect on the choice of mode for lighter clusters.

Again, lets look at the example of decaying $u\bar{u}$ and $d\bar{d}$ clusters with a cluster mass just above threshold for a $\pi^+\pi^-$ decay.

$$W(\pi^0, \pi^0|u\bar{u}) = P_u w^2_{\pi^0} s^2_{\pi^0} p^*_{\pi^0,\pi^0}, \tag{3.21}$$

$$W(\pi^0, \pi^0|d\bar{d}) = P_d w^2_{\pi^0} s^2_{\pi^0} p^*_{\pi^0,\pi^0}, \tag{3.22}$$

$$W(\pi^+,\pi^-|u\bar{u}) = P_d w_{\pi^+} w_{\pi^-} s_{\pi^+} s_{\pi^-} p^*_{\pi^+,\pi^-}, \tag{3.23}$$

$$W(\pi^+,\pi^-|d\bar{d}) = P_u w_{\pi^+} w_{\pi^-} s_{\pi^+} s_{\pi^-} p^*_{\pi^+,\pi^-}. \tag{3.24}$$

With the same assumptions as before ($s_i = 1, p^*_{\pi^0,\pi^0} = p^*_{\pi+,\pi^-}, P_u = P_d$) we get

$$P(\pi^0,\pi^0|u\bar{u}) = P(\pi^0,\pi^0|d\bar{d}) = \frac{w^2_{\pi^0}}{w^2_{\pi^0} + w_{\pi^+} w_{\pi^-}}, \tag{3.25}$$

$$P(\pi^+,\pi^-|u\bar{u}) = P(\pi^+,\pi^-|d\bar{d}) = \frac{w_{\pi^+} w_{\pi^-}}{w^2_{\pi^0} + w_{\pi^+} w_{\pi^-}}. \tag{3.26}$$

Finally this gives us the ratio

$$\pi^0 : \pi^+ : \pi^- = w^2_{\pi^0} : \frac{1}{2} w_{\pi^+} w_{\pi^-} : \frac{1}{2} w_{\pi^+} w_{\pi^-}, \tag{3.27}$$

and as the w_i's are static *a priori* weights, we can choose them to be $w_{\pi^0} = \frac{1}{\sqrt{2}}\sqrt{w_{\pi^+} w_{\pi^-}}$ which will give the correct ratio of $1:1:1$.

Now if we allow the cluster mass to increase so that the $\pi^0\eta$ mode is accessible, we get the same weights for the $\pi^0\pi^0$ and the $\pi^+\pi^-$ modes and the new $\pi^0\eta$ mode has a weight of

$$W(\pi^0,\eta|u\bar{u}) = P_u w_{\pi^0} w_\eta s_{\pi^0} s_\eta p^*_{\pi^0,\eta}, \tag{3.28}$$

and similarly for the $d\bar{d}$ cluster. The sum of the weights, without some common factors and using the same assumptions, is

$$\Sigma = w^2_{\pi^0} + 2 w_{\pi^0} w_\eta \frac{p^*_{\pi^0,\eta}}{p^*_{\pi^0,\pi^0}} + w_{\pi^+} w_{\pi^-}. \tag{3.29}$$

This is the same for both the $u\bar{u}$ and $d\bar{d}$ clusters. Therefore the probabilities are now

$$P(\pi^0,\pi^0) = \frac{w^2_{\pi^0}}{\Sigma}, \tag{3.30}$$

$$P(\pi^0,\eta) = \frac{2 w_{\pi^0} w_\eta}{\Sigma} \frac{p^*_{\pi^0,\eta}}{p^*_{\pi^0,\pi^0}}, \tag{3.31}$$

$$P(\pi^+, \pi^-) \quad = \quad \frac{w_{\pi^+} w_{\pi^-}}{\Sigma}. \tag{3.32}$$

To find the ratio we need to make sure we count the particles correctly. Doing so we find

$$\pi^0 : \pi^+ : \pi^- = 2w_{\pi^0}^2 + 2w_{\pi^0} w_\eta \frac{p^*_{\pi^0,\eta}}{p^*_{\pi^0,\pi^0}} : w_{\pi^+} w_{\pi^-} : w_{\pi^+} w_{\pi^-}. \tag{3.33}$$

After forcing the right ratio near the $\pi^+\pi^-$ threshold by setting the w_i's, we now don't have the right ratio after the $\pi^0\eta$ threshold.

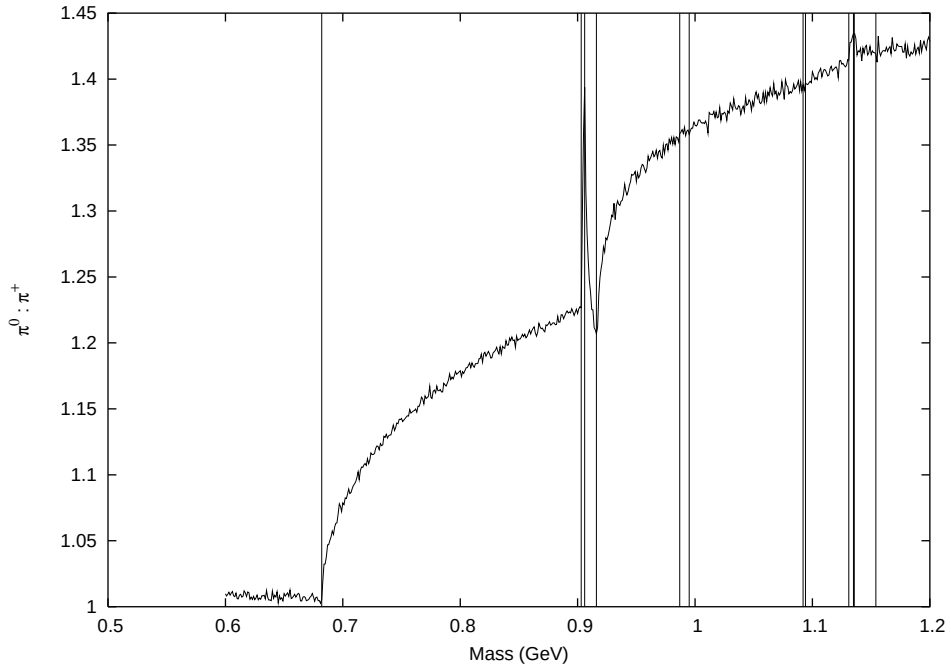


Figure 3.5: The ratio of π^0 to π^+ as the cluster mass is increased. When new hadrons become accessible the ratios differ from one, sometimes drastically.

Figure 3.5 shows the ratio of $\pi^0 : \pi^+$ as the cluster mass is increased. After a new production threshold is reached the ratio changes. The change can become quite dramatic and the average ratio between thresholds differs from unity.

Though the ratios vary as the cluster mass changes, this seems overall to be an improvement from the original cluster decays. We no longer have the problem that the addition of new hadrons to the lists causes decreased probability. We now have

increased the probability of choosing a flavour content when new hadrons of that content are added. In doing so, however, we have created a new problem. In baryon-conserving processes, we only have clusters with a quark–antiquark pair, rather then some diquark–quark combination. This means two baryons must be created at a time by drawing a diquark–antidiquark pair out of the vacuum. Doing so requires a cluster with a high mass, which in turn means a new meson added to the list is likely to be available. Consider a $u\bar{u}$ cluster with high mass. If we have 5 hadrons with $u\bar{d}$ flavour and 2 baryons with uud flavour we will have only 4 possible combinations of the uud flavour baryons but 25 modes of the $u\bar{d}$ flavour. If all the modes are accessible this makes the baryon probabilities lower, as they are all normalized by the sum of all accessible modes. If we then add a 6th hadron to the $u\bar{d}$ list, we now have 36 combinations. If, again, these are all accessible that decreases the probability of creating a baryon even more. This problem is addressed and solved in the new method implemented in Herwig++ and described in the next section.

3.3 Herwig++

The nonperturbative splitting, cluster formation and cluster fission described in section 3.1 has been implemented in Herwig++. The method for decaying clusters is similar to the Kupco method described in section 3.2.2 but has been changed to account for the lack of baryon production. Results of the hadronization process in Herwig++ are given in this section. Also the changes to the cluster decay model are presented here.

The concept of colour preconfinement says that the mass distribution of the colour singlet systems, after the parton shower, are independent of the centre-of-mass energy of the hard process. Figure 3.1 shows the cluster mass distribution generated from Herwig++ using the default parameter set. It can be seen that this distribution is

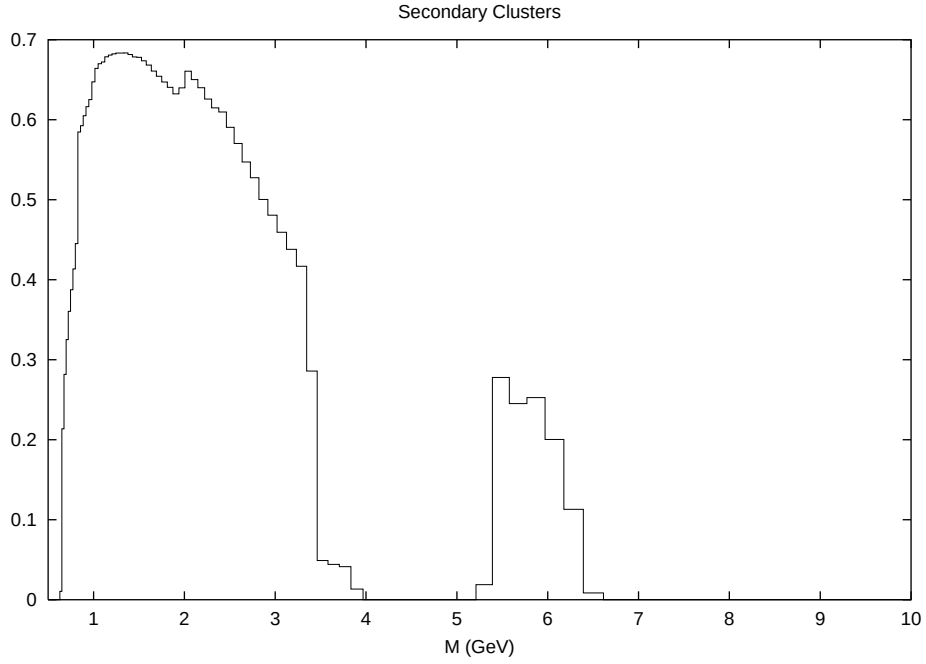


Figure 3.6: The cluster mass distribution after cluster fission. This is using the default parameter set. It can be seen that the heavy clusters no longer exist and are instead folded into the lighter part of the distribution.

indeed independent of Q, the c.m. energy of the hard process.

The cluster fission stage described in section 3.1.3 truncates this distribution by breaking apart clusters that are above the threshold in (3.1). Figure 3.6 shows the new distribution after this process. The two bumps around 2 GeV and 5.5 GeV are from the charm and bottom clusters. This figure is for the default parameter set ($Cl_{pow} = 2.0, Cl_{max} = 3.2$ GeV). As can be seen, the heavier part of the distribution is folded into the lighter part.

3.3.1 Herwig++ Cluster Decay Algorithm

The algorithm used in Herwig++ for cluster decays is very similar to the method proposed by Kupco. The weights of a particular decay mode is given by (3.19). As discussed previously, the problem with Kupco's method is that there are not enough baryons pro-

Parameter	Herwig++	Old Model
δ	2.3 GeV	2.3 GeV
m_g	0.750 GeV	0.750 GeV
α_s(1GeV)	0.630882 GeV	0.630882 GeV
$\texttt{Cl}_{\texttt{max}}$	3.2 GeV	3.0 GeV
$\texttt{Cl}_{\texttt{pow}}$	2.0 GeV	2.0 GeV
PSplt1	1	1
PSplt2	0.33	0.33
B1Lim	0.0	0.0
ClDir1	1	1
ClDir2	1	1
ClSmr1	0.40	0.0
ClSmr2	0.0	0.0
$\texttt{Pwt}_d$	1.0	1.0
$\texttt{Pwt}_u$	1.0	1.0
$\texttt{Pwt}_s$	0.85	1.0
$\texttt{Pwt}_c$	1.0	1.0
$\texttt{Pwt}_b$	1.0	1.0
$\texttt{Pwt}_{qq}$	0.55	0.65
Singlet Weight	1.0	1.0
Decuplet Weight	0.7	1.0

Table 3.1: The parameters for Herwig++. The first group are shower parameters, the second are all of the hadronization parameters. The meaning of all the parameters is given in chapter 4.

duced. This is because of the quantity of possible decay modes in the meson sector compared to the quantity of modes in the baryon sector. To fix this problem a way to separate the two sectors was needed.

The new algorithm separates the meson sector from the baryon sector. The weights from (3.19) are still used, but the sum of weights used for normalizing is only summed over modes in either the meson or baryon sector. If a cluster mass is high enough to decay into the lightest baryon pair then the parameter $\texttt{Pwt}_{qq}$ is used to decide whether to use the baryon sector or meson sector. There is a $(1+\texttt{Pwt}_{qq})^{-1}$ probability of using the meson sector and a $\frac{\texttt{Pwt}_{qq}}{1+\texttt{Pwt}_{qq}}$ probability of using the baryon sector. This change not only allows for more baryons to be created but also gives more direct control of how many baryons are produced through the diquark parameter $\texttt{Pwt}_{qq}$.

3.3.2 Results

In this section the results of the different hadronization schemes of Herwig++ for e^+e^- events at 91.2 GeV are presented. The values of the parameters in Herwig++ used for this study are given in table 3.1. This section presents only a few results related to the new hadronization method. Full results of Herwig++ including event shapes and jet physics are given in chapter 5.

Table 3.2 shows the results of the new cluster hadronization algorithm in comparison to the old algorithm. The column labeled 'Old Model' is the result of using the old algorithm with the new parton shower in Herwig++. The column Herwig++ is the result of using the new cluster hadronization model with the new parton shower variables. The last column, labeled Fortran, is the data generated using the Fortran HERWIG program, version 6.5. The data in this table are combined and updated from a variety of sources, see ref. [58].

Neither of the Herwig++ implementations in the table have been tuned but the results in the Fortran column are the result of tuning. 10000 events were used for the 'Old Model' data and the Herwig++ results are for 100000 events. As the models are different a new set of parameters are needed. The parameters for this are also shown in table 3.1. The results given in the last column, Fortran, are taken from [59].

As we can see from the multiplicity results Herwig++ is able to obtain multiplicities that are as good as, if not better than, the old HERWIG 6.5 results. The χ^2 of the data sets are given in table 3.3.

We also want to make sure that the momentum distributions of the hadrons match those from the data. x_p is defined as

$$x_p = \frac{2\,|\vec{p}|}{E_{max}}, \tag{3.34}$$

Particle	Experiment	Measured	Old Model	Herwig++	Fortran
All Charged	M,A,D,L,O	20.924 ± 0.117	20.22*	20.814	20.532*
γ	A,O	21.27 ± 0.6	23.032	22.6745	20.74
π^0	A,D,L,O	9.59 ± 0.33	10.273	10.0777	9.88
$\rho(770)^0$	A,D	1.295 ± 0.125	1.235	1.31596	1.07
$\pi^\pm$	A,O	17.04 ± 0.25	16.297	16.9461	16.74
$\rho(770)^\pm$	O	2.4 ± 0.43	1.994	2.14345	2.06
η	A,L,O	0.956 ± 0.049	0.886	0.892915	0.669*
$\omega(782)$	A,L,O	1.083 ± 0.088	0.859	0.91576	1.044
$\eta'(958)$	A,L,O	0.152 ± 0.03	0.13	0.135935	0.106
K^0	S,A,D,L,O	2.027 ± 0.025	2.121*	2.06214	2.026
$K^*(892)^0$	A,D,O	0.761 ± 0.032	0.667	0.681661	0.583*
$K^*(1430)^0$	D,O	0.106 ± 0.06	0.065	0.078918	0.072
$K^\pm$	A,D,O	2.319 ± 0.079	2.335	2.28551	2.250
$K^*(892)^\pm$	A,D,O	0.731 ± 0.058	0.637	0.657485	0.578
$\phi(1020)$	A,D,O	0.097 ± 0.007	0.107	0.113829	0.134*
p	A,D,O	0.991 ± 0.054	0.981	0.946807	1.027
Δ^{++}	D,O	0.088 ± 0.034	0.185	0.092217	0.209*
Σ^-	O	0.083 ± 0.011	0.063	0.071454	0.071
Λ	A,D,L,O	0.373 ± 0.008	0.325*	0.384086	0.347*
Σ^0	A,D,O	0.074 ± 0.009	0.078	0.091162	0.063
Σ^+	O	0.099 ± 0.015	0.067	0.077027	0.088
$\Sigma(1385)^\pm$	A,D,O	0.0471 ± 0.0046	0.057	0.031159*	0.061*
Ξ^-	A,D,O	0.0262 ± 0.001	0.024	0.028565	0.029
$\Xi(1530)^0$	A,D,O	0.0058 ± 0.001	0.014*	0.007782	0.009*
Ω^-	A,D,O	0.00125 ± 0.00024	0.001	0.001439	0.0009
$f_2(1270)$	D,L,O	0.168 ± 0.021	0.113	0.150273	0.173
$f_2'(1525)$	D	0.02 ± 0.008	0.003	0.011739	0.012
$D^\pm$	A,D,O	0.184 ± 0.018	0.322*	0.318519*	0.283*
$D^*(2010)^\pm$	A,D,O	0.182 ± 0.009	0.168	0.180251	0.151*
D^0	A,D,O	0.473 ± 0.026	0.625*	0.570354*	0.501
$D_s^\pm$	A,O	0.129 ± 0.013	0.218*	0.194775*	0.127
$D_s^{*\pm}$	O	0.096 ± 0.046	0.082	0.066209	0.043
J/Ψ	A,D,L,O	0.00544 ± 0.00029	0.006	0.003605*	0.002*
Λ_c^+	D,O	0.077 ± 0.016	0.006*	0.022621*	0.001*
$\Psi'(3685)$	D,L,O	0.00229 ± 0.00041	0.001*	0.001775	0.0008*

Table 3.2: Multiplicities per event at 91.2 GeV. We show results from Herwig++ with the implementation of the old cluster hadronization model (Old Model) and the new model (Herwig++), and from HERWIG 6.5 shower and hadronization (Fortran). Experiments are ALEPH(A), DELPHI(D), L3(L), OPAL(O), MK2(M) and SLD(S). The $*$ indicates a prediction that differs from the measured value by more than three standard deviations.

	Old Model	Herwig++	HERWIG 6.5
All Data from Table 3.2	$\chi^2 = 543.84/35 = 15.54$	$\chi^2 = 277.16/35 = 7.92$	$\chi^2 = 490.52/35 = 14.01$

Table 3.3: χ^2 results for the different cluster decay methods.

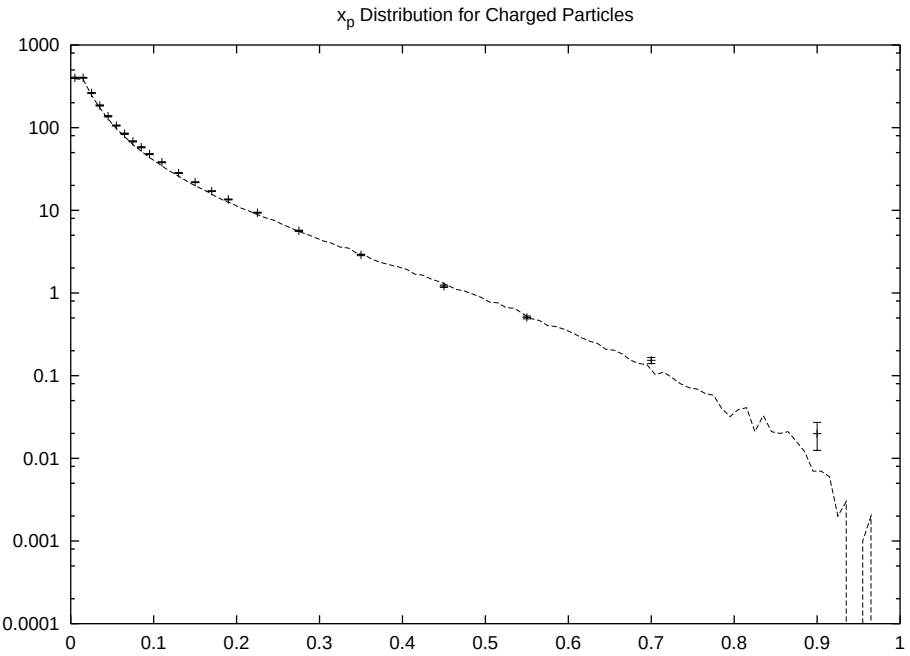


Figure 3.7: Plot of x_p for all flavours. Data from OPAL collaboration.

where E_{max} is the centre-of-mass energy. Presented here are the results for x_p for all charged particles and $\xi_p^{uds} = \ln \frac{1}{x_p^{uds}}$ for all charged particles in uds events. Also the momentum distributions of $\pi^{\pm}$, $K^{\pm}$ and $p^{\pm}$ are shown. All results are compared to data from the OPAL collaboration [60,61]. Again we can see that the results from Herwig++ are in good agreement with the data. More detailed results are given in chapter 5.

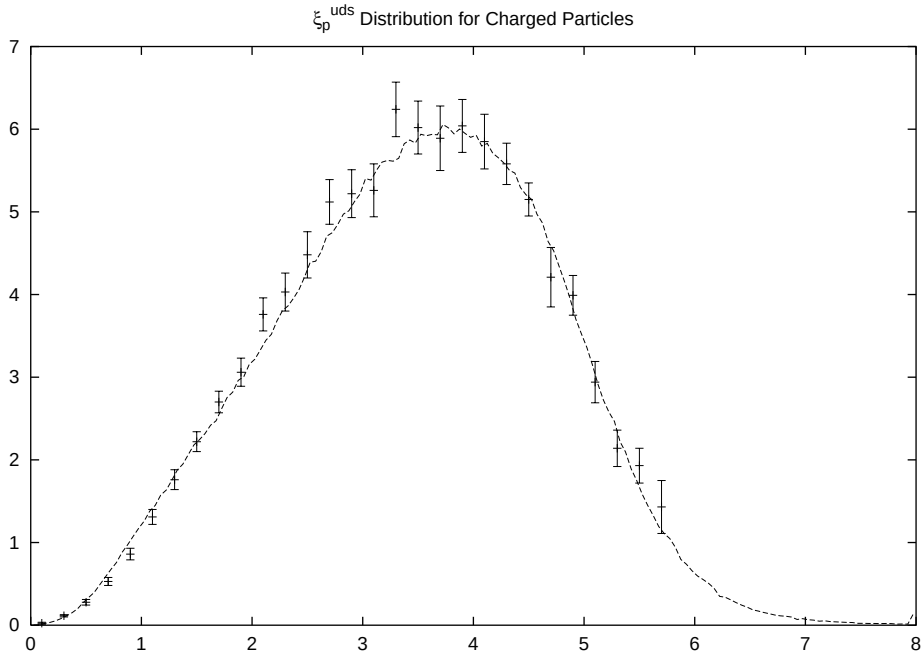


Figure 3.8: Plot of $\ln 1/x_p$ for uds flavours. Data from OPAL collaboration.

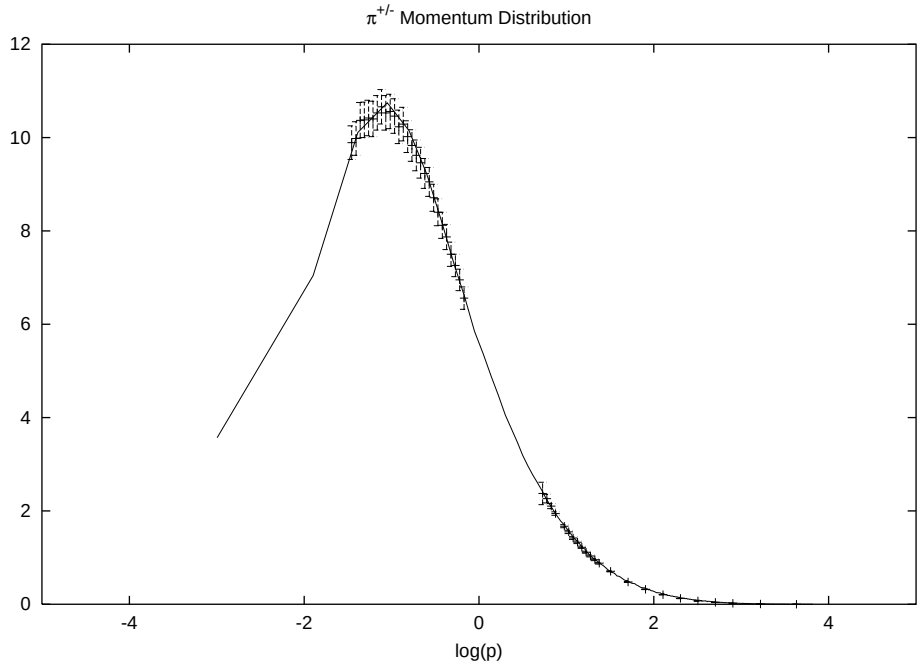


Figure 3.9: Plot of $\log p$ for $\pi^{\pm}$. Data from OPAL collaboration.

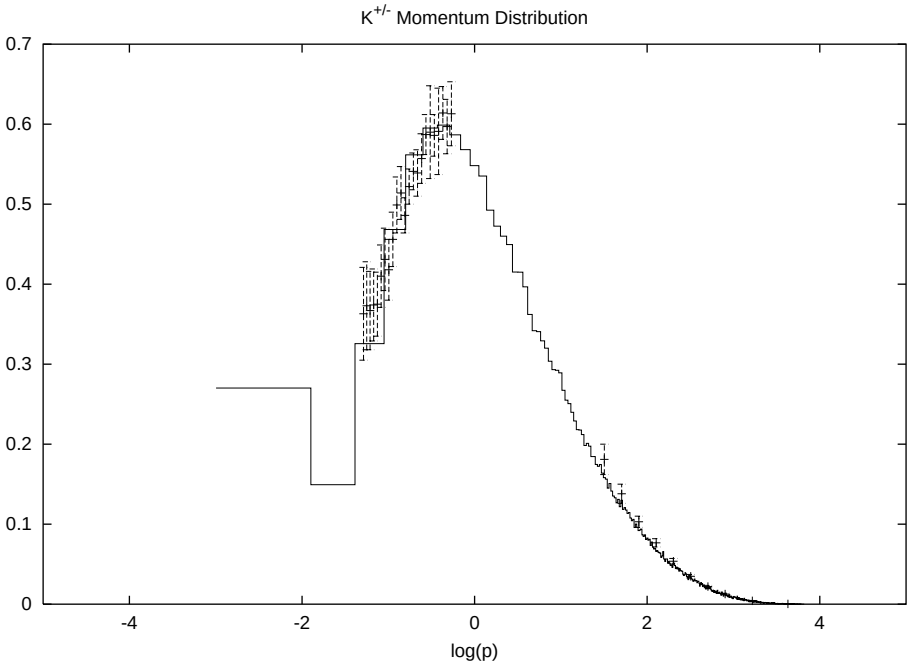


Figure 3.10: Plot of $\log p$ for $K^{\pm}$. Data from OPAL collaboration.

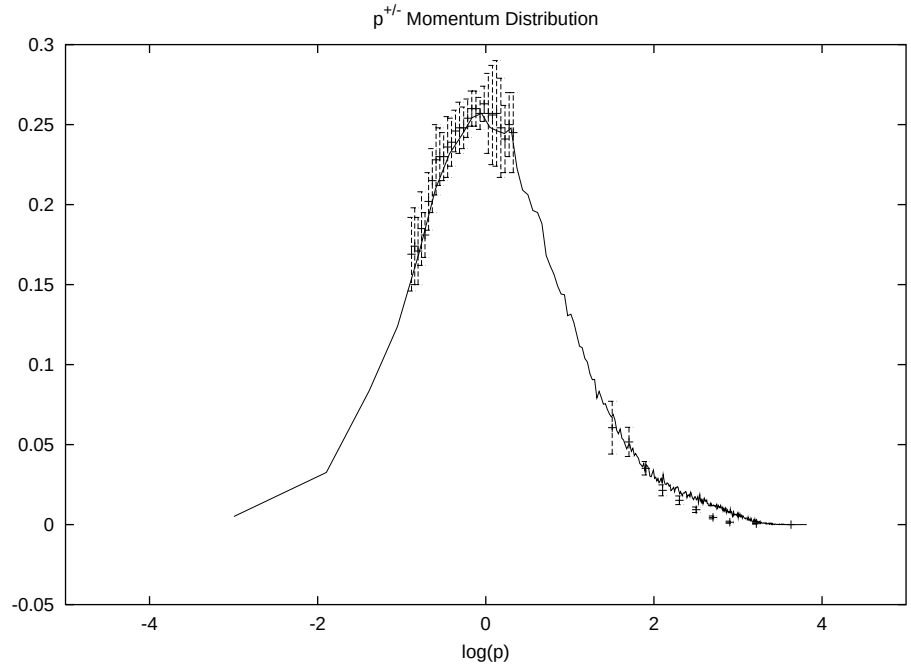


Figure 3.11: Plot of $\log p$ for $p^{\pm}$. Data from OPAL collaboration.

Chapter 4

Herwig++

The new generation of high energy colliders such as the Large Hadron Collider (LHC) or a future linear collider (NLC) require new tools for the simulation of signals and backgrounds. The widely used event generators HERWIG [62] and PYTHIA [25] underwent tremendous development during the LEP era and have reached the limit of reasonable maintenance. Therefore these programs (Pythia7) [63] as well as new projects, like SHERPA [32], are being completely (re-)developed in the object-oriented programming language C++.

Chapters 2 and 3 introduced two new theoretical improvements to the original HERWIG [24] Monte Carlo event generator. In this chapter I present the implementation of these two improvements in the new Herwig++ [41] event generator as well as the implementation of the various other parts of the event generator. The Herwig++ event generator is built on top of ThePEG [64]. ThePEG is an administrative library which defines tools and data structures which are commonly used by Monte Carlo event generators. ThePEG and Herwig++ also use CLHEP [65]. This is a package that provides general HEP functions.

This chapter will contain some discussions about object-oriented programming. As

this thesis is not intended to be a discussion of object-oriented programming and design, detailed discussion of these matters is given only as a reference [66]. For the purposes of this chapter a few keywords and very simple conceptual definitions of these terms are given here. Again, this is not intended as a complete description, just a guide for understanding the text of this chapter.

Classes are the main components of an object-oriented program. A very simplistic view of a class is as a function or algorithm (with accompanying data). In a sense a class just serves to implement some functionality or apply some algorithm. A class is much more complex and diverse than this, but for the current purposes this should suffice.

A class can be a particular type of another class. For example, a class could define how a particle behaves. It could contain all the data as well as some special functions, such as boosts, that are useful when working with a particle. During the shower more information is needed for a particle, such as the Sudakov basis quantities α and β in (2.1). Rather then completely redefining a ShowerParticle to behave almost identically to a regular Particle, except for the new data, we can instead *inherit* the old Particle class into the new ShowerParticle class. This means that all of the original data and functionality of the Particle class is also present in the ShowerParticle class, but we can now add the new data and functions for the ShowerParticle. This is useful because in the code we can just pass all the Particles and ShowerParticles around together as one set of Particles. Then if we want to perform special functions on the ShowerParticles we can identify which of the Particles is really a ShowerParticle and apply our operations.

A function in a class can be defined as a *virtual* function. Classes that inherit a class with a virtual function can redefine that function. These functions can actually be defined to be non-existent. If this is done the class is known as an *abstract* class. This allows for the definition of an *interface* without defining the implementation. For example, there are many things that all matrix elements must have; all matrix elements

must have a set of incoming and outgoing particles. But the actual values of the matrix elements, for a given set of incoming and outgoing particles with momenta, are different. This is a perfect candidate for abstraction. A class `ME` can be defined that provides the definition of a function `double me2(incoming,outgoing)` but does not define it. Then any class that inherits `ME` can implement the function. If we now think about an algorithm that uses a matrix element, such as computing a cross section, we can define this algorithm independent of what matrix element we want to use. It just needs to know that the matrix element class will return a double when given a set of incoming and outgoing particles (or momenta).

This brief discussion of a few terms from object-oriented programming should be adequate to comprehend the rest of this thesis. The rest of this chapter describes the implementation of the various parts of the Monte Carlo event generator: hard subprocess, PDFs, parton shower, underlying event, hadronization, and decays.

4.1 Hard Subprocess

As discussed in the first chapter, there is only a small set of matrix elements currently implemented into Herwig++. For e^+e^- annihilation there is just one. It is the $e^+e^- \to \gamma^*/Z^0 \to q\bar{q}$ matrix element. Though this is not itself a QCD matrix element it inherits the `ME2to2QCD` class from ThePEG. This is because the `ME2to2QCD` class has a function which returns the number of accessible flavours for a given scale, which is needed for the matrix element at a given scale.

There is a much larger set of matrix elements for pp collisions. All of these inherit the `ME2to2QCD` class. Provided with ThePEG is the set of matrix elements: `MEqq2qq`, `MEQG2QG`, `MEGG2GG`, `MEQQ2QQ`, `MEGG2QQ`, `MEQQ2qq`, `MEQQ2GG` and `MEQq2Qq`. The lower and upper case q are used to decipher processes that have different flavours. Also imple-

mented with Herwig++ is a Drell-Yan matrix element $q\bar{q} \to \gamma^*/Z^0 \to \ell^+\ell^-$. This type of process is particularly useful for studying the initial-state shower.

Implementation of the hard subprocess requires both a matrix element and a phase space sampler. The method of sampling phase space is important for the efficiency of the Monte Carlo. As discussed before, matrix elements[a] can have several peaks and valleys. Using a uniform sampling for this is extremely inefficient and advanced samplers can improve the efficiency drastically. The default sampler used in Herwig++ is known as the ACDC sampler. This is a component of ThePEG. ACDC is an acronym of Auto-Compensating Divide-and-Conquer Phase Space Generator [67]. This algorithm uses a divide-and-conquer scheme to divide the phase space into uniform sections which have different maxima. Figure 4.1 shows an example of a function which has been divided into two sections, each of which is sampled uniformly.

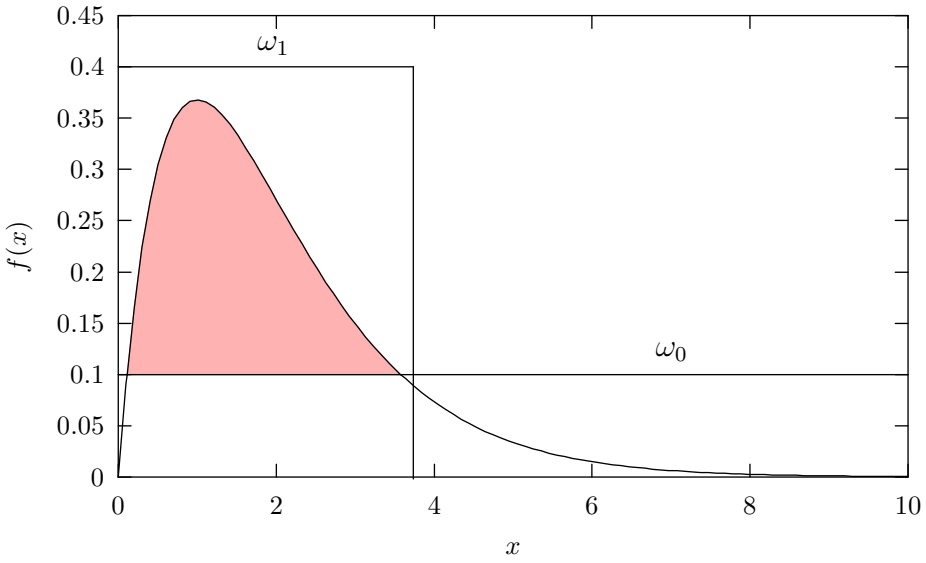


Figure 4.1: This figure shows a function being integrated which has been divided into two regions, ω_0 and ω_1.

The way the algorithm works is to generate points uniformly in phase-space. When a point is generated in which the matrix element is larger than the value being sampled

[a]By 'matrix element' we actually mean 'square modulus of the matrix element'

under, ω_0, the space is divided into two regions. A region is found in which the function is larger than ω_0 and this region is then sampled under ω_1. The auto-compensating part of the the algorithm compensates for the fact that the peak has been undersampled and oversamples the new region until it is consistent. The shaded region of fig. 4.1 shows this new region which is auto-compensated.

4.2 PDF

Chapter 1 introduced the concept of the parton distribution functions. As mentioned earlier there are different ways to parameterize the non-perturbative component of the PDF. Two such implementations are the Glück-Reya-Vogt (GRV) [68] PDFs and the Martin-Roberts-Stirling-Thorne (MRST) [69] PDFs. Both of these have been implemented in Herwig++.

The implementation of the PDFs requires the implementation of four virtual functions inherited from the `PDFBase` class. These functions are:

- `bool canHandleParticle(tcPDPtr particle) const`; this function is used to specify what hadrons this PDF can work with.
- `cPDVector partons(tcPDPtr particle) const`; this function is used to specify what partons can be extracted from the given hadron.
- `ApproxMap approx(tcPDPtr particle, const PDFCuts &) const`; this function is used to specify, approximately, the upper limits of the parton densities, of each parton, for the given hadron. The `PDFCuts` class is used to give as input the kinematical region that the PDF will be used in.
- `double xfl(tcPDPtr particle, tcPDPtr parton, Energy2 partonScale, double l, Energy2 particleScale = 0.0*GeV2) const`; this function is the actual im-

plementation of the PDF. This function returns the momentum fraction as $l = \log(1/x)$.

Optionally, the PDF can also be specified with the functions:

- `double xfx(tcPDPtr particle, tcPDPtr parton, Energy2 partonScale, double x, double eps = 0.0, Energy2 particleScale = 0.0*GeV2) const;`
  this function is used to specify the PDF and return the momentum fraction as x. By default, this just returns $\exp(-\texttt{xfl}(\dots))$.
- `double xfvl(tcPDPtr particle, tcPDPtr parton, Energy2 partonScale, double l, Energy2 particleScale = 0.0*GeV2) const;`
  this function returns just the valence part of the PDF with momentum fraction $l = \log(1/x)$. By default this just returns 0.
- `double xfvx(tcPDPtr particle, tcPDPtr parton, Energy2 partonScale, double x, double eps = 0.0, Energy2 particleScale = 0.0*GeV2) const;`
  this function is the same as `xfvl` except it returns the momentum fraction as x instead of l. It also returns 0 by default.

The GRV PDFs have been implemented as part of ThePEG. The GRV94L and GRV94M PDFs have been implemented. These are two implementations of the GRV method where optimal fits of the distributions of the partons have been made to different data.

The MRST PDFs have also been implemented as part of Herwig++. This implementation is a wrapper to a previous C++ implementation [70]. The original implementation was able to read in the data from a file. The files are standardized and when new data is analyzed and integrated into the distribution, new data files are created. The new

versions are easily integrated into the event generator by simply reading in these data files.

In order for the event generator to function correctly something must be done with the remnant of the incoming hadron left behind when a parton interacts. These are handled by `RemnantHandlers`. ThePEG has a default remnant handler, `BaryonRemnant`, but this is designed to be integrated with the Pythia7 string fragmentation model. This remnant handler works in two ways. If a valence parton is chosen by the PDF then the only remnant is the colour connected diquark remaining in the beam particle. If a sea parton is used in the hard subprocess then a colour connected parton is produced along with a colourless hadron. The implementation of `BaryonRemnant` in ThePEG requires a $p_\perp$-generator and a z-generator. Though the z-generator can be designed to work with other hadronization models, it is really a feature of the string fragmentation model and isn't something that is needed in Herwig++.

Instead, since Herwig++ uses a cluster hadronization model, we use a different remnant handler. This remnant handler, also called `BaryonRemnant`, is really nothing more than a place-holder in Herwig++; the initial-state shower in Herwig++ is designed to evolve back to a valence parton. This class simply generates a diquark, with either spin 1 or spin 0, with a flavour depending on the parton drawn by the PDF. This remnant is often wrong because the initial-state shower evolves backwards to a valence parton but the generation of this valence parton is not known at the time the remnant is created. Correctly handling the remnant is instead implemented as part of the backwards evolution and the original remnant created by `BaryonRemnant` is replaced[b]

[b] `BaryonRemnant` is only implemented as it is a requirement of ThePEG.

4.3 Parton Shower

In chapter 1 I have developed the theoretical background of the parton shower and how it takes the matrix elements calculated perturbatively and evolves down to a stage where hadronization models are valid. Chapter 2 presented the development of a set of evolution variables used in the Herwig++ shower. In this section I present the implementation specific features of the Herwig++ shower. This includes how the shower is initialized, how it terminates and how the momenta of the particles are set.

4.3.1 Hard Matrix Element Corrections

Given an n–jet process it is sometimes possible to calculate the matrix element for $n+1$ jets and this matrix element can be used in the Monte Carlo. There are already soft and collinear jets of order $n+1$ from the shower of the n-jet matrix element and these jets must be matched with the $n+1$-jet matrix element to avoid double counting. This matching is known as *hard matrix element corrections*.

Hard matrix element corrections have only been implemented for the process $e^+e^- \to \gamma^*/Z^0 \to q\bar{q}$. To make a hard correction a pair of three-body phase space variables, $x, \bar{x}$ are generated according to the original two-jet matrix element. Emissions in the dead region of fig. 2.3 are only accepted according to the three-jet matrix element. In the case of $e^+e^- \to q\bar{q}$ only 3% of the events are corrected by the hard matrix element.

If a hard emission is added the $q\bar{q}$-final state is replaced by a $q\bar{q}g$-final state and the orientation of either the quark or the anti-quark is kept with weights x^2 and $\bar{x}^2$, respectively. This results in properly oriented three-jet events, apart from finite mass effects [71]. This procedure takes into account the most important subleading higher-order corrections that are not given by the parton shower.

The `ShowerVariables` class has a variable, `MECorrMode`, which sets whether the hard matrix elements are on or off.

4.3.2 Initial Conditions

As we saw in chapter 1 there is an angular ordering principle that restrict the showering to occur only in the cone of half angle between two colour connected partons. Since the shower is restricted to a cone in relation to one of its colour partners, the first step of the shower is to determine which colour partners the soft and collinear showers will occur between. There is a flag, `Approach`, used in the `PartnerFinder` class that sets the way the shower partner is chosen. If this flag is zero, then the partner is set completely randomly amongst the partons that are colour-connected to it and the partners of all partons are set independently of each other. This means if we have particle a, which chooses its shower partner to be particle b, particle b does not have to choose a as its shower partner. Instead, if it had colour partners a and c it would randomly choose between these. On the other hand, if `Approach` is set to 1 then it randomly selects a shower partner and sets both particles to be shower partners with each other.

A partner is chosen for each gauge 'charge' of the parton. For example, a quark has a QED charge and a QCD charge. A colour partner and an electric charge partner are both set. This allows the QED showers to compete directly with QCD ones.

As shown in chapter 2, the evolution of a particle is carried out in the Sudakov basis,

$$q = \alpha p + \beta n + q_\perp, \tag{4.1}$$

where p is the momentum of the particle which is evolving and n is a lightlike ($n^2 = 0$) vector with 3-momentum in the 'backwards' direction, which is conventionally set to that of the colour partner of the particle in their c.m. frame. $q_\perp$ is in the transverse

direction and satisfies $q_\perp \cdot p = q_\perp \cdot n = 0$.

Once the partner is chosen the initial value of the evolution variable, $\tilde{q}$, is set. The value of this variable depends on whether both partners are initial-state, final-state or a combination. If one parton is initial-state and one is final-state the values are

$$\tilde{q}_i = \sqrt{(p_i + p_f)^2 + m_f^2}, \tag{4.2}$$

$$\tilde{q}_f = \sqrt{(p_i + p_f)^2 + 2m_f^2}, \tag{4.3}$$

where p_i is the momentum of the initial-state parton and p_f is the momentum of the final-state parton. This corresponds to (2.107) for the most symmetrical choice, where $\tilde{q}^2 = \tilde{\kappa} Q^2$. For final-final shower partners the initial conditions are

$$\tilde{q}_1 = \sqrt{(p_1 + p_2) \cdot (p_1 + n_2)}, \tag{4.4}$$

$$\tilde{q}_2 = \sqrt{(p_1 + p_2) \cdot (p_2 + n_1)}, \tag{4.5}$$

where p_1 and p_2 are the momenta of the incoming partons. n_1 has three momentum equal to $\boldsymbol{p}_1$ in the c.m. frame and $n_1^2 = 0$; similarly for n_2. This corresponds to (2.54) for the symmetric choice. Lastly, for the choice of two initial-state particles the initial conditions are

$$\tilde{q}_1 = \tilde{q}_2 = \sqrt{(p_1 + p_2)^2}. \tag{4.6}$$

This corresponds to the symmetric case of $\tilde{\kappa}_1 = \tilde{\kappa}_2 = 1$, shown in fig. 2.8.

4.3.3 Initial-State Shower

The initial-state shower evolution begins with the two incoming partons that have been chosen from the PDF. These partons are considered as on-shell partons in the hard matrix element calculation and the initial $\tilde{q}$ are set as described in the previous section.

Once the initial value is set for the evolution variable, each parton then evolves independently of each other. The evolution of one parton proceeds using the *veto* algorithm. For each possible type of branching, $a \to bc$, a new $\tilde{q}$ and a z are generated based on ratios of the Sudakov form factor

$$R^i_{ba}\left(\tilde{q}_i, \tilde{q}_{i+1}\right) = \frac{f_b(x/z, \tilde{q})\Delta_{ba}\left(\tilde{q}_C, \tilde{q}_i\right)}{f_b(x, \tilde{q})\Delta_{ba}\left(\tilde{q}_C, \tilde{q}_{i+1}\right)}, \tag{4.7}$$

with f_b the PDF and Δ_{ba} the Sudakov form factor

$$\Delta_{ba}\left(\tilde{q}_C, \tilde{q}\right) = \exp\left\{-\int_{\tilde{q}_C}^{\tilde{q}} \frac{d\tilde{q}^2}{\tilde{q}^2}\int dz \frac{\alpha_{\mathrm{S}}\left(z, \tilde{q}\right)}{2\pi} P_{ba}\left(z, \tilde{q}\right)\Theta\left(\boldsymbol{p}_\perp > 0\right)\right\}. \tag{4.8}$$

$\tilde{q}_C$ is the lower cutoff and by default is set to be the non-perturbative gluon mass, $m_g = 750$ MeV. The running coupling, $\alpha_{\mathrm{S}}(z, \tilde{q})$, depends on the evolution scale and the momentum fraction. The argument is $(1-z)^2\tilde{q}^2$ for reasons given in chapter 2. The P_{ba} are the quasi-collinear splitting functions for the massive partons [72]. For QCD branchings these are

$$\begin{aligned}
P_{qq}(z, \tilde{q}) &= C_F\left[\frac{1+z^2}{1-z} - \frac{2m_a^2}{z(1-z)\tilde{q}^2}\right], & (4.9)\\
P_{qg}(z, \tilde{q}) &= T_R\left[1 - 2z(1-z) + \frac{2m_a^2}{z(1-z)\tilde{q}^2}\right], & (4.10)\\
P_{gq}(z, \tilde{q}) &= C_F\left[\frac{1+(1-z)^2}{z} - \frac{2m_a^2}{z(1-z)\tilde{q}^2}\right], & (4.11)\\
P_{gg}(z, \tilde{q}) &= C_A\left[\frac{z}{1-z} + \frac{1-z}{z} + z(1-z)\right], & (4.12)
\end{aligned}$$

and $m_a = 0$ for initial-state radiation. For the QED case, we change α_{S} to the fine structure constant α_{em} and use the branching for $q \to q\gamma$. Ignoring the parton mass this is

$$P^\gamma_{qq}(z, \tilde{q}) = e_a^2\frac{1+z^2}{1-z}, \tag{4.13}$$

where e_a^2 is the electric charge of the parton, in units of elementary charge.

The $\Theta(\boldsymbol{p}_\perp > 0)$ function in (4.8) is used to ensure that it is possible to reconstruct transverse momentum, $\boldsymbol{p}_\perp$, from the evolution variables. $\tilde{q}$ determines the relative transverse momentum. For quark branching this is

$$|\boldsymbol{p}_{\perp i}| = \sqrt{(1-z_i)^2\tilde{q}_i^2 - z_i Q_g^2}. \tag{4.14}$$

For the gluon branching, in the initial state shower, this is

$$|\boldsymbol{p}_{\perp i}| = \sqrt{z^2(1-z)^2\tilde{q}^2 - Q_g^2}. \tag{4.15}$$

Eqn. (4.7) gives the probability of no branching above the scale $\tilde{q}_{i+1}$. $1-R^i_{ba}(\tilde{q}_i, \tilde{q}_{i+1})$ is therefore the probability for the next branching to happen above $\tilde{q}_{i+1}$. The derivative with respect to $\tilde{q}_{i+1}$ is then the probability density for the next branching to happen at the scale $\tilde{q}_{i+1}$.

Since (4.7) is not directly solvable, the veto algorithm is used. In this algorithm each part of the distribution is sampled independently by a function which is always greater than the desired one, and a veto is placed on the emission if the ratio of the actual function to the approximated function is larger then some random number. This gives several veto points

$$w_1 = \Theta(\boldsymbol{p}_\perp > 0), \quad w_2 = \frac{P_{ba}(z,\tilde{q})}{g_{ba}(z)}, \quad w_3 = \frac{\alpha(z,\tilde{q})}{\alpha_{\max}}, \quad w_4 = \frac{f_b(x/z,\tilde{q})}{f_b(x,\tilde{q})}, \tag{4.16}$$

where

$$g_{qq}(z) = \frac{2C_F}{1-z}, \tag{4.17}$$

$$g_{qg}(z) = T_R, \tag{4.18}$$

$$g_{gq}(z) = \frac{2C_F}{z}, \tag{4.19}$$

$$g_{gg}(z) = C_A \left[\frac{1}{1-z} + \frac{1}{z} \right], \tag{4.20}$$

$$g^{\gamma}_{qq}(z) = \frac{2e_a^2}{1-z}. \tag{4.21}$$

The trick to the veto algorithm, to ensure that things are correctly sampled, is that first a $\tilde{q}$ is generated according to

$$\tilde{q}^2 = \tilde{q}_s^2 \exp\left(\frac{2\pi}{\alpha_{\max}\left(I(z_+) - I(z_-)\right)} \ln \mathcal{R} \right), \tag{4.22}$$

with $\tilde{q}_s$ the current scale, z_+ and z_- the upper and lower bounds on z, respectively and $I(z_0)$ the value of g_{ba} integrated over z and from zero to z_0. A z is then generated according to the approximate functions, g_{ba} between z_+ and z_-. Each veto is then tested. If a veto fails a new $\tilde{q}$ is generated. This time, however, instead of $\tilde{q}_s$ in (4.22) it is equal to the $\tilde{q}$ that was vetoed. If this $\tilde{q}$ is smaller than $\tilde{q}_C$ then there is no more branchings. The largest value of $\tilde{q}$ generated from each of the branchings decides which of the branchings to use.

Once a splitting has been chosen a new initial state parton and a new final state parton are created. The final state parton is taken as an on-shell parton. This will be put off-shell during the final-state evolution. The momentum fraction, z, from the Sudakov form factor and the new scale $\tilde{q}$ are passed to the new initial state parton so it can split. This process is repeated until no new scale is chosen below $\tilde{q}_C$ for any of the branchings. From the interpretation of the Sudakov form factor, this means that there is no more branching and the evolution of this parton has terminated.

As discussed before, beam remnants aren't handled properly by the remnant handler. Instead they are handled here. When the initial-state evolution of a parton terminates a set of forced branchings are imposed. There are three types of termination points: a valence quark, a gluon, or a sea quark. If the evolution terminates on a valence quark,

then a diquark[c] of the appropriate flavours is produced as the beam remnant. If instead the evolution terminates on a gluon, a forced splitting of the gluon from a quark of a valence flavour is imposed. The momentum fraction z is distributed according to the splitting function. The $\tilde{q}$ is just distributed by $dP = d\tilde{q}/\tilde{q}$ between $\tilde{q}_C$ and $\tilde{q}_s$. The remnant is again just a (di)quark of the flavour(s) that remain in the beam particle. The most complex case is when the evolution terminates on a sea quark. In this case two forced splittings are imposed. The first is to force a splitting into a gluon. Again the momentum fraction is distributed according to the appropriate splitting function. The $\tilde{q}$ is also distributed as $dP = d\tilde{q}/\tilde{q}$ between $\tilde{q}_C$ and $\tilde{q}_s$. The new gluon is then forced to split into a valence quark in the same manner described above.

This process is repeated for both incoming partons. Once they have evolved and their remnants are correctly set the momenta of all the partons needs to be set. The initial condition is that the beam particles are coming in with known momenta. The Sudakov variable α of the first parton is then set to unity, the β is set to zero and its x is known from the backwards evolution. Each child of the parton then has its α set to $z\alpha_i$, for the initial state particle and $(1-z)\alpha_i$ for the final state partner produced during the backward evolution. The β for the on-shell final state partons is

$$\beta'_{i+1} = \frac{m^2 + \boldsymbol{p}'^2_{\perp i+1}}{2\alpha_i p \cdot n}. \tag{4.23}$$

The p and n vectors are the Sudakov basis vectors for the shower and

$$\boldsymbol{p}'_{\perp i+1} = z\boldsymbol{p}_{\perp i} - \boldsymbol{p}_\phi \sqrt{z^2\tilde{q}^2 - (1-z)Q_g^2}. \tag{4.24}$$

Here $\boldsymbol{p}_\phi$ is a 2 vector given by $(\cos\phi, \sin\phi)$. By default ϕ is distributed uniformly in the region $[0, 2\pi]$ but improvements, such as the spin correlations described in chapter 1,

[c] If a beam particle was a meson this would just be a (anti-)quark

can be implemented. The values for the initial-state parton $i+1$ are

$$\begin{aligned} \beta_{i+1} &= \beta_i - \beta'_{i+1}, & (4.25)\\ \boldsymbol{p}_{\perp i+1} &= \boldsymbol{p}_{\perp i} - \boldsymbol{p}'_{\perp i+1}. & (4.26) \end{aligned}$$

All of the α's and $\beta's$ are set until the parton involved in the hard subprocess is reached. This fully defines the momentum of the partons. Unfortunately, this does not guarantee momentum conservation at the hard subprocess. Instead the momentum of the partons must be "shuffled" in order to impose momentum conservation. Rescaling the momentum of the partons in the hard subprocess, and correspondingly boosting all the partons involved in the evolution of each initial-state partons, allows for momentum conservation. This rescaling can affect other properties, however, and we want to constrain the value of the rescaling so that certain properties are retained.

In proton-proton collisions we want to conserve the rapidity, Y, and the c.m. energy squared, M^2, of the hard process, while rescaling each incoming parton independently. Each parton has its momentum shifted by k_1 and k_2, given in the Sudakov base by

$$\begin{aligned} q_1 &= k_1\alpha_1 p_1 + \frac{\beta_1}{k_1} n_2 + p_{1\perp},\\ q_2 &= \frac{\alpha_2}{k_2} p_2 + k_2\beta_2 n_1 + p_{2\perp}. & (4.27) \end{aligned}$$

These shifts, k_1 and k_2, are applied so that the virtuality of the partons is conserved. Conserving rapidity and the c.m. energy squared requires

$$\begin{aligned} \frac{\beta_1}{k_1} + k_2\beta_2 &= k_1\alpha_1 + \frac{\alpha_2}{k_2}, & (4.28)\\ 1 + \frac{(p_{1\perp} + p_{2\perp})^2}{M^2} &= \left(k_1\alpha_1 + \frac{\alpha_2}{k_2}\right)\left(k_2\beta_2 + \frac{\beta_1}{k_1}\right). & (4.29) \end{aligned}$$

Both of these give a quadratic solution for each k value. This leads to four different

combinations of values which must be considered. Two combinations of these four have negative k's. This corresponds to flipping the event, e.g. the parton which started from the $-z$ direction ends up on the $+z$ direction and vice versa. In some cases the reconstruction of the shower fails. In these cases the shower is *vetoed* and started again from the on-shell incoming partons.

4.3.4 Final-State Shower

The final-state shower is similar to the initial state shower. The partons start as on-shell partons but are taken off shell by the evolution. The initial scale of a parton is set according to its shower partner and the partons evolve down to the scale $\tilde{q}_C$ in the same manner as the initial-state shower, except the probability of emission is modified. Instead of R^i_{ba} we have

$$R^f_{ba} = \frac{\Delta_{ba}(\tilde{q}_C, \tilde{q}_i)}{\Delta_{ba}(\tilde{q}_C, \tilde{q}_{i+1})}. \tag{4.30}$$

This means that the extra veto, w_4 from (4.16), is not applied to final-state branchings. Though the generation of the branchings is similar, the kinematics of the final-state shower is different from that of the initial-state shower. From the results in chapter 2 we find the relative transverse momentum as

$$|\boldsymbol{p}_{\perp i}| = \sqrt{(1-z_i)^2 z_i^2(\tilde{q}_i^2 - \mu^2) - z_i Q_g^2}, \tag{4.31}$$

$$|\boldsymbol{p}_{\perp i}| = \sqrt{z_i^2(1-z_i)^2\tilde{q}_i^2 - \mu^2}, \tag{4.32}$$

for quark branching and gluon branching respectively and $\mu = \max(m_a, Q_g)$.

All of the partons created during the evolution also shower starting at the $\tilde{q}$ that they were produced at. Every parton showers until the condition of no more branching is met. Once this is reached the parton is put on its mass shell so that momentum reconstruction can be done.

The partons produced in the hard subprocess have their α set to unity and their β set to 0. Each of children then have their Sudakov variable α set to $z\alpha_i$ or $(1-z)\alpha_i$, set by the convention of the splitting function. The β for the parton corresponding to the z momentum fraction is

$$\beta_{i+1} = \frac{q^2_{i+1} + \boldsymbol{q}^2_{\perp i+1} - \alpha^2_{i+1}p^2}{2\alpha_{i+1}p\cdot n}, \tag{4.33}$$

where p and n are the Sudakov basis vectors of the shower. The β corresponding to the $(1-z)$ momentum fraction is just

$$\beta'_{i+1} = \beta_i - \beta_{i+1}. \tag{4.34}$$

Since the final partons are on their mass shells, the initial partons aren't. This means instead of having $p_j^2 = m_j^2$ they have acquired some virtuality, $p_j^2 = q_j^2$. The original momenta, $p_j = \left(\sqrt{\boldsymbol{p}_j^2 + m_j^2}, \boldsymbol{p}_j\right)$ in the centre-of-mass frame, define some properties of the hard subprocess. In the case of e^+e^- annihilation we want to preserve the centre-of-mass energy

$$\sqrt{s} = \sum_{j=1}^{n} \sqrt{m_j^2 + \boldsymbol{p}_j^2} \tag{4.35}$$

while keeping the sum of momenta equal to zero. This requires momentum reshuffling. To conserve $\sqrt{s}$ we rescale the momentum of each jet by a common factor, k, that is determined from the equation

$$\sqrt{s} = \sum_{j=1}^{n} \sqrt{q_j^2 + k\boldsymbol{p}_j^2}. \tag{4.36}$$

This effectively creates a Lorentz transformation which is applied to all partons in the final state. For $n = 2$ outgoing particles from the hard process (4.36) can be solved for k explicitly. For $n > 2$ this is done numerically.

4.3.5 Soft Matrix Element Corrections

We saw earlier that a hard correction is added in to populate the 'dead region' of the $(n+1)$–body matrix element. The parton shower can also be improved in the shower regions of phase space by restricting relatively hard gluons that are produced in the shower. These gluons are no longer in the domain of of validity of the quasi-collinear approximation.

Each of the $\boldsymbol{p}_\perp$ values that are generated during the evolution of one jet is tracked. If a new $\boldsymbol{p}_\perp$ is larger than any previously generated during the evolution a soft matrix element correction is applied to it [73]. This correction assumes all other emissions are infinitely soft and we can treat the emission as part of a three-body phase space (e.g. $q\bar{q}g$). This allows us to compute the three-body variables, $(x, \bar{x})$, from the parton shower variables $(\tilde{q}, z)$ and the respective Jacobian. The ratio of the hard matrix element and the shower is then compared to a random number and the emission is vetoed if the ratio is smaller. This requires that the shower approximation is larger than the matrix element everywhere in phase space. Otherwise the ratio must have some factor applied to ensure that the ratio is always less than 1. This has been studied for several cases in chapter 2 and the relevant ratios have been derived.

4.3.6 Parameterization of Q_g

The cutoff Q_g is introduced to regularize the soft gluon singularities in the splitting functions. The relative transverse momentum, $\boldsymbol{p}_\perp$ is related to the Sudakov variables of the parton branching by

$$\boldsymbol{p}_\perp = \boldsymbol{q}_{\perp i+1} - z\boldsymbol{q}_{\perp i}. \tag{4.37}$$

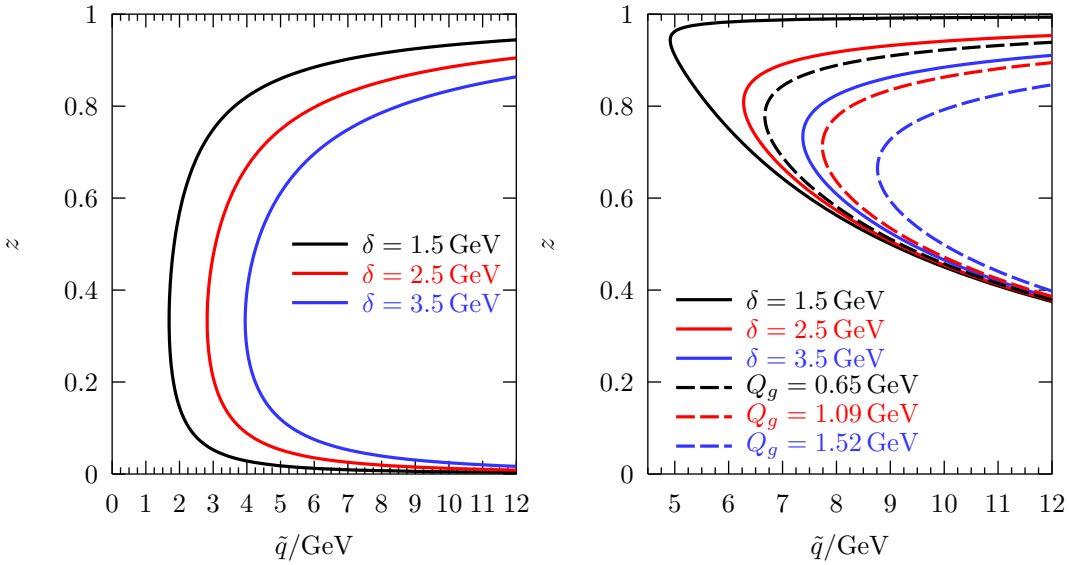


Figure 4.2: Available phase space of light (left) and b–quarks (right) for $q \to qg$ splitting for various values of Q_g and depending on the parameterization in terms of δ, eq. (4.40). The dashed lines on the right correspond to a Q_g which is not mass dependent, obtained by setting $m = 0$ in (4.40).

z is required to correspond to a real value of $\boldsymbol{p}_\perp$. For a gluon splitting this is explicitly

$$z_- < z < z_+, \qquad z_\pm = \frac{1}{2}\left(1 \pm \sqrt{1 - \frac{4\mu}{\tilde{q}}}\right), \tag{4.38}$$

with $\tilde{q} > 4\mu$. For quark splittings z is the solution of a cubic but is always in the range

$$\frac{\mu}{\tilde{q}} < z < 1 - \frac{Q_g}{\tilde{q}}. \tag{4.39}$$

This allows z to be generated within these regions and simply rejected if it lies outside phase space.

Q_g is parameterized according to

$$Q_g = \frac{\delta - 0.3m}{2.3}, \tag{4.40}$$

where δ is the parameter `cutoffKinScale` in the class `ShowerVariables` and m is the

mass of the parton splitting. This cutoff is used to give the gluons a minimum virtuality which ensures that they are able be put on a mass shell with non-zero mass. As we can see from figure 4.2, the form of this function is also to ensure that b quarks don't have an extra artificial cutoff that over-restricts the phase space of splitting b quarks.

4.4 Hadronization

Chapter 3 described in detail the hadronization scheme used in Herwig++. This section will discuss the features of the code and the implementation of the specific features. Table 4.1 is a table with all the relevant parameters of the hadronization.

The main driver of the code is the `ClusterHadronization` class. This class organizes and directs which classes are used next. This class also handles the unusual case of a hadronization veto. This can occur during the light cluster reshuffling, which will be described below. In order to direct the algorithm this class has references to `PartonSplitter`, `ClusterFinder`, `ColourReconnector`, `ClusterFissioner`, and the two decayer classes `LightClusterDecayer` and `ClusterDecayer` classes. The class `ColourReconnector` by default doesn't do anything. This class is designed to allow for a different colour configuration in the forming of the clusters, for example this is done in SHERPA [32].

The algorithm begins by taking the gluons, which are on a mass-shell, and non-perturbatively splitting these into $q\bar{q}$ pairs. The possible flavours depend on what the mass of the gluon is. This is done for all final-state gluons produced in the final-state shower (including the final-state evolution of the gluons produced during the initial-state shower). The splitting is done by the class `PartonSplitter`. This class has a reference to the `GlobalParameters` object that defines all the global parameters used throughout Herwig++.

Parameter	Class	Description	Default Value
$\texttt{Cl}_{\texttt{max}}$	ClusterFissioner	The maximum cluster mass before a cluster fissions.	3.2
$\texttt{Cl}_{\texttt{pow}}$	ClusterFissioner	The cluster mass exponent that controls cluster fissions.	2.0
PSplt1	ClusterFissioner	This is the mass splitting parameter for $udsc$ flavours.	1.0
PSplt2	ClusterFissioner	This is the mass splitting parameter for b quarks.	1.0
B1Lim	LightClusterDecayer	Parameter to set the limit of light b clusters before forcing decay to one hadron.	0.0
ClDir1	ClusterDecayer	Flag to turn on or off the the smearing for non-b quarks.	1
ClDir2	ClusterDecayer	Flag to turn on or off the the smearing for b quarks.	1
ClSmr1	ClusterDecayer	Gaussian smearing of non-b quarks	0.0
ClSmr2	ClusterDecayer	Gaussian smearing of b quarks	0.0
$\texttt{Pwt}_d$	HadronSelector	Weight of d quarks	1.0
$\texttt{Pwt}_u$	HadronSelector	Weight of u quarks	1.0
$\texttt{Pwt}_s$	HadronSelector	Weight of s quarks	1.0
$\texttt{Pwt}_c$	HadronSelector	Weight of c quarks	1.0
$\texttt{Pwt}_b$	HadronSelector	Weight of b quarks	1.0
$\texttt{Pwt}_{qq}$	HadronSelector	Weight of diquarks	1.0
SngWt	HadronSelector	Weight of baryon singlets	1.0
DecWt	HadronSelector	Weight of baron decuplets	1.0
DKMode	HadronSelector	Which hadron decay method to use	2

Table 4.1: This is a table of all of the relevant hadronization parameters. Most of the parameters are discussed in chapter 3 and the rest are discussed in this section.

Once the gluons have all been split into $q\bar{q}$ pairs clusters are formed out of the colour connected partons. This is done by the `ClusterFinder` class. This class simply searches all coloured partons and creates a colour singlet state out of the colour connected partons. There is also a feature to work with partons that are created by colour sources or sinks which stem from baryon violating processes. This feature has not been tested though and this needs to be done once baryon violating processes are included into Herwig++. After the clusters are formed, if a `ColourReconnector` class was defined to change things, it would be called to do so at this stage.

After the `ClusterFinder` class has created all of the colour singlet clusters, they are passed to the `ClusterFissioner` class. As was described in chapter 3, the heavy clusters are decayed into lighter clusters. This occurs in three ways. If the mass drawn for one of the new clusters in not heavy enough to form the lightest possible pair of hadrons, given the set of flavours, the decay is instead forced to $C \to H + C'$ where H is a hadron and C' is a new cluster. H is the lightest hadron of the flavour of the cluster whose mass is too light. The mass generated for the light cluster is changed to equal the mass of the lightest hadron. If the mass of the cluster C' produced in the decay would then violate the phase space bounds it is set to have the largest mass available. The next special case is when both clusters are too light to form a hadron pair. This decay, $C \to H_1 + H_2$, follows the same procedure as the previous one. Each hadron is the lightest hadron of the appropriate flavours and the masses are set to the mass of the respective hadron. If this violates the phase space bounds, then the cluster isn't decayed via this mechanism. In fact, if this still violates phase space it will be handled by the `LightClusterDecayer` later. The last possible way for a heavy cluster to be decayed is directly into two new clusters, $C \to C'_1 + C'_2$. In any case, if any of the new clusters is still too heavy, it also decays via this algorithm.

After the heavy clusters have been handled, the clusters which are too light are

treated. This is done by the `LightClusterDecayer`. This takes clusters which are too light to decay into a pair of hadrons and turns them directly into a single hadron. It may be possible that a cluster has formed which is even too light to decay directly into the lightest hadron. In this case the momentum of this cluster is 'reshuffled' with a cluster which is nearby in x-space. The algorithm only 'borrows' momentum from a given cluster once during the process. If it is impossible to find enough momentum to borrow then the event is vetoed and begun again from the `ClusterFissioner` stage. In the rare case that after vetoing several times there is still not a possible configuration to decay all the clusters then the whole event is vetoed.

The last step of the hadronization is turning the clusters into hadrons. This is done by the `ClusterDecayer` class. This has a reference to a `HadronSelector` class. The `HadronSelector` class has been implemented with a flag, `DKMode`, which is used to select which of the methods discussed in chapter 3 to use. By default this flag is set to 2, which is the new method. If it was changed to 0 the implementation of the HERWIG method would be used. A value of 1 is for the Kupco method. Once a pair of hadrons has been selected by one of the methods then the cluster is decayed. The decay products are generated in the direction of the constituent quark. This direction can have a Gaussian smearing applied to it to. If the flag `ClDir` is set and `ClSmr` is larger then 0.0001, then cluster decay is in the direction

$$\cos\theta = \cos\theta_q \cos\theta_{\rm smr} - \sin\theta_q \sin\theta_{\rm smr}, \tag{4.41}$$

where θ_q is the θ of the original constituent quark and

$$\cos\theta_{\rm smr} = 1 + \texttt{ClSmr} \log\mathcal{R}, \tag{4.42}$$

with $\mathcal{R}$ a random number. The azimuthal angle is distributed uniformly in $[-\pi, \pi]$.

4.5 Decays

In chapter 1 the general scheme of the decays was presented. Since the algorithms implemented in Herwig++ are very basic, in this section I will explain the way to setup the decay modes in Herwig++ and how one would implement a new set of decay modes.

In Herwig++ the default decays are defined in the file `Hw64Decays.in`. In this file all of the decay modes are included. An entry of a decay mode takes the form

`decaymode` a->$b, c, d, \ldots$; BR 0/1 Class

for the decay $a \to b+c+d+\ldots$, BR is the branching ratio and the 0/1 option specifies whether to turn this mode off or on, respectively. The last argument specifies which class to use to perform the decay. By default Herwig++ has only the `HwDecayer` class, the `HeavyDecayer` class and the `QuarkoniumDecayer` class. All of these classes use a parameter `MECode` to specify the matrix element code to use.

The class `HwDecayer` is the most common class used in the default Herwig++ decays. This just implements the decays as described in chapter 1 based on the `MECode` parameter. A value of 100 uses the free massive $(V-A)(V-A)$ matrix element, for example $\tau^- \to \bar{\nu}_e e^- \nu_\tau$. 101 is the code to use the bound massive matrix element. This would be used for a decay such as $K^- \to \bar{\nu}_e e^- \pi^0$. Any other value of the parameter uses an isotropic n-body decay.

The class `HeavyDecayer` takes heavy mesons and decays the heavy parton weakly. The W produced in the decay is also decayed all in one step. This produces 4 partons, two from the decay of the W, one from the production of the W and the remaining spectator parton. For example a chain could look like, $B^0 \to d\bar{c}W^+ \to d\bar{c}\bar{d}u$. The decay products have the colour connections set properly and the administrative class

`HwDecayHandler` directs the program to reshower and rehadronize these partons.

The last decayer, the `QuarkoniumDecayer`, is designed to work with heavy mesons and baryons that decay into gluons and quarks. Possible modes are to 3 gluon states, 2 gluons and a photon, just 2 gluons or a $q\bar{q}$ state. This class handles a `MECode` of 130 by using a positronium matrix element. Decays from this class also lead to showering and hadronization of the partonic decay products. An example of this type of decay is $\eta_c \to gg$.

Implementation of a new decay mode requires the definition of only two functions. A new decay matrix element must inherit the `Decayer` class from ThePEG. The two functions that must be implemented are

- `bool accept(const DecayMode &) const`; this function is used to indicate if a particular decay mode can be handled. For example if you implemented a decay matrix element for $\tau^{\pm}$ decays, if the incoming particle was not a $\tau^{\pm}$, this function would simply return false.
- `ParticleVector decay(const DecayMode &, const Particle &) const`; this function returns the decay products with their momentum set. The input is the decay mode selected based on the branching ratios, and the instance of the particle which is decaying.

Improving the decay modes of Herwig++ is currently underway. This is a lengthy process and as long as experimentalists study the decays of particles, improving the decay mode matrix elements, as well as the branching ratios, will always be possible. But as will be shown in the next chapter, even with the simple decay matrix elements, fits to particle spectra are good.

Chapter 5

Results

5.1 Introduction

This chapter presents results from the Herwig++ . The results given here are for e^+e^- annihilation events, as this is the first step in the redevelopment of HERWIG. In order to have full control of the basic physics steps that are simulated, it was thought to be very important to put the new generator on a firm basis with respect to LEP and SLC results before upgrading it to be able to deal with initial-state showers and the other requirements for the simulation of lepton-hadron and hadron-hadron collisions. Therefore, thorough tests have been performed on the predictions of the generator against a wide range of observables that have been measured at LEP and SLC. We have also explored the sensitivity to the most important parameters and cutoffs. These results are not the result of a high-precision tuning: the main aim here is rather to show the results of the program and that it is able to give results as acceptable as those generated by its predecessor HERWIG for a reasonable choice of parameters.

5.1.1 Main features of the code

As was discussed in detail in the last three chapters, the main stages of the simulation of events is the same as in HERWIG [62]. However, in comparison to its predecessor, Herwig++ features a new parton shower and an improved cluster hadronization model. At present, hadronic decays are implemented in the same fashion as they were in HERWIG.

As discussed in chapter 4, the program is based on the Toolkit for High Energy Physics Event Generation (ThePEG) [64] and the Class Library for High Energy Physics (CLHEP) [65]. They are utilized in order to take advantage of the extended general functionality they can provide. The usage of ThePEG unifies the event generation framework with that of Pythia7. This will provide benefits for the user, as the user interface, event storage etc. will appear to be the same. The implementations of the physics models, however, are completely different and independent from each other.

The simulation of e^+e^- annihilation events starts with an initial hard process $e^+e^- \rightarrow \gamma^*/Z^0 \rightarrow q\bar{q} + \gamma\gamma$. The final state photons simulate QED radiation from the initial state, so that a radiative return can be properly simulated. For these results we are only interested in the details of the QCD parton shower in the final state. The final-state parton shower starts with a quark and antiquark that carry momenta p_q and $p_{\bar{q}}$, respectively, and have an invariant mass squared of $Q^2 = (p_q + p_{\bar{q}})^2$. For the e^+e^- results, the only detail we are concerned with in relation to initial-state radiation is that the centre-of-mass frame of the $q\bar{q}$–pair is slightly boosted with respect to the collider laboratory frame and that Q may be different from the e^+e^- centre-of-mass energy. We have made sure that the applied cuts on the energy of the annihilating e^+e^- subsystem are the same as those used in the experimental analyses.

Currently, proton-proton collisions are being studied. These results are simulated by

starting with the Drell-Yan hard process $q\bar{q} \to \gamma^*/Z^0 \to \ell^+\ell^-$, with $\ell^\pm$ a charged lepton. These results concentrate on the effects of the initial-state radiation from the incoming quarks and gluons.

5.1.1.1 Parton shower

As has been shown in the preceding chapters, the partonic evolution from the large scale of the hard collision process down to hadronic scales via the coherent emission of partons, mainly gluons, is simulated on the basis of the Sudakov form factor. Starting from the hard process scale Q_0, subsequent emissions at scales Q_i and momentum fractions z_i are randomly generated as a Markov chain on the basis of the soft and collinear approximation to partonic matrix elements. Chapter 2 has shown that for Herwig++ we have chosen a new framework of variables, generically called $(\tilde{q}, z)$. Here, $\tilde{q}$ is a scale that appears naturally in the quasi-collinear approximation of massive partonic matrix elements and generalizes the evolution variable of HERWIG to the evolution of massive quarks. z is a relative momentum fraction; the evolution is carried out in terms of the Sudakov decomposition of momenta in the frame where the respective colour partners are back-to-back. As in HERWIG, the use of the new variables allows for an inherent angular ordering of the parton cascade, which simulates coherence effects in soft gluon emission. The details of this underlying formalism have been described in chapter 2.

The most important parameter of the parton shower that we will be concerned with in this chapter is the cutoff parameter Q_g, which regularizes the soft gluon singularity in the splitting functions and determines the termination of the parton shower. This is set by δ in (4.40). Less important but relevant in extreme cases is the treatment of the strong coupling constant at low scales. We have parametrized $\alpha_S(Q)$ below a small scale $Q_{\min} > \Lambda_{QCD}$ in different ways. We keep $Q_{\min}$ generally to be of the order of 1 GeV, where we expect non-perturbative effects to become relevant. Below that scale $\alpha_S(Q)$

can optionally be

- set to zero, $\alpha_S(Q < Q_{\min}) = 0$,
- frozen, $\alpha_S(Q < Q_{\min}) = \alpha_S(Q_{\min})$,
- linearly interpolated in Q, between 0 and $\alpha_S(Q_{\min})$,
- quadratically interpolated in Q, between 0 and $\alpha_S(Q_{\min})$.

We put the final partons of the shower evolution on their constituent mass shells, since the non-perturbative cluster hadronization will take over at this scale, so we usually have kinematical constraints that keep Q above $Q_{\min}$, in which case the treatment below $Q_{\min}$ is irrelevant. Typically, $\alpha_S(Q_{\min}) \sim 1$ here.

5.1.1.2 Hadronization and decay

As discussed in chapter 3, the partonic final state is turned into a hadronic final state within the framework of the cluster hadronization model of HERWIG [74]. All three methods of cluster decays have been implemented in Herwig++, but the new cluster hadronization model is used for the results given in this chapter. The emerging hadrons are possibly unstable and eventually decay. The decay matrix elements and modes correspond to those in HERWIG.

5.2 e^+e^- Annihilation

This section presents the results for e^+e^- annihilation events. The properties of different measurements are discussed and the comparison of Herwig++ to data is presented. Histograms for all the distributions have been booked in the same bins as the experimental data. For a given bin i we then compare the data D_i value with the Herwig++

Parameter	Herwig++	Initial
$\alpha_s(M_Z)$	0.118	0.114
δ/GeV	2.3	—
m_g/GeV	0.750	—
$Q_{\min}$/GeV in $\alpha_s(Q_{\min})$	0.631	—
$\texttt{Cl}_{\texttt{max}}$/GeV	3.2	3.35
$\texttt{Cl}_{\texttt{pow}}$	2.0	—
PSplt1	1	—
PSplt2	0.33	—
B1Lim	0.0	—
ClDir1	1	—
ClDir2	1	—
ClSmr1	0.40	—
ClSmr2	0.0	—
Pwt_d	1.0	—
Pwt_u	1.0	—
Pwt_s	0.85	1.0
Pwt_c	1.0	—
Pwt_b	1.0	—
Pwt_{qq}	0.55	1.0
Singlet Weight	1.0	—
Decuplet Weight	0.7	1.0

Table 5.1: The parameters for Herwig++ used in this study. The first group are shower parameters, the second are all of the hadronization parameters. In the third column we show initial values of our study, taken from HERWIG.

Monte Carlo result M_i. Given the data errors δD_i (statistical plus systematic, added in quadrature) and Monte Carlo errors δM_i (statistical only), we can calculate a χ^2 for each observable. We keep the statistical error of the Monte Carlo generally smaller than the experimental error. In distributions where the normalization is not fixed, such as momentum spectra, we allow the normalization of the Monte Carlo to be free to minimize χ^2. The normalization is then tested separately against the average multiplicity. In all other cases we normalize histograms to unity.

As we do not want to put too much emphasis on a single observable or a particular region in phase space where the data are very precise, in computing χ^2 we set the relative experimental error in each bin to $\max(\delta D_i/D_i, 5\%)$. This takes into account the fact that the Monte Carlo is only an approximation to QCD and agreement with the data within 5% would be entirely satisfactory. The general trend for the preferred range of a single parameter was however never altered by this procedure.

After normalization the ratio

$$R_i = \frac{M_i - D_i}{D_i} \pm \left(\frac{\delta M_i}{D_i} \oplus \frac{M_i \delta D_i}{D_i^2} \right) \tag{5.1}$$

is computed for each bin in order to see precisely where the model fails. This ratio as well as the relative experimental error and the relative contribution of each bin to the χ^2 of an observable is plotted below each histogram.

5.2.1 Strategy

We have taken χ^2 values for hadron multiplicities into account in the same way as we weighted the event shapes. In general the multiplicities of individual particle species are sensitive to a completely different set of parameters. The general strategy was to get a good value for the total number of charged particles with a reasonable set of values for

the parton shower cutoff parameter δ and the maximum cluster mass parameter $\texttt{Cl}_{\texttt{max}}$. Once this was fixed, the hadronization parameters that determine the multiplicities of individual particle species were determined. Following this we compared this 'preferred' set of parameters with the 'default' set from HERWIG. The resulting parameter set is shown in Table 5.1.

A wide range of observables have been tested in order to study the aspects of the model. Event shape variables and multiplicities are considered in order to test the dynamical aspects of the parton shower and hadronization models, which are closely linked at their interface by the parton shower cutoff parameter, δ. Ideally, the models should combine smoothly at scales where $Q_g \sim 1$ GeV. Many of the figures shown in this chapter contain three sets of plots per figure. In order from top to bottom these are

- the actual distribution. The Herwig++ result is plotted as a histogram together with the experimental data points;
- the ratio R_i (5.1) together with an error band showing the relative statistical and systematic errors;
- the relative contribution of each data point to the total χ^2 of each plot.

5.2.2 Hadron multiplicities

The charged particle multiplicity distribution and the overall multiplicities of a wide range of hadron species have been taken to test the overall flow of quantum numbers through the different stages of the simulation. This also allows a thorough test of the new hadronization model, developed in chapter 3, against the measured observables.

Table 3.2 showed the results of the new cluster hadronization algorithm in comparison to the old algorithm. Even before systematic tuning, we can see that the overall results

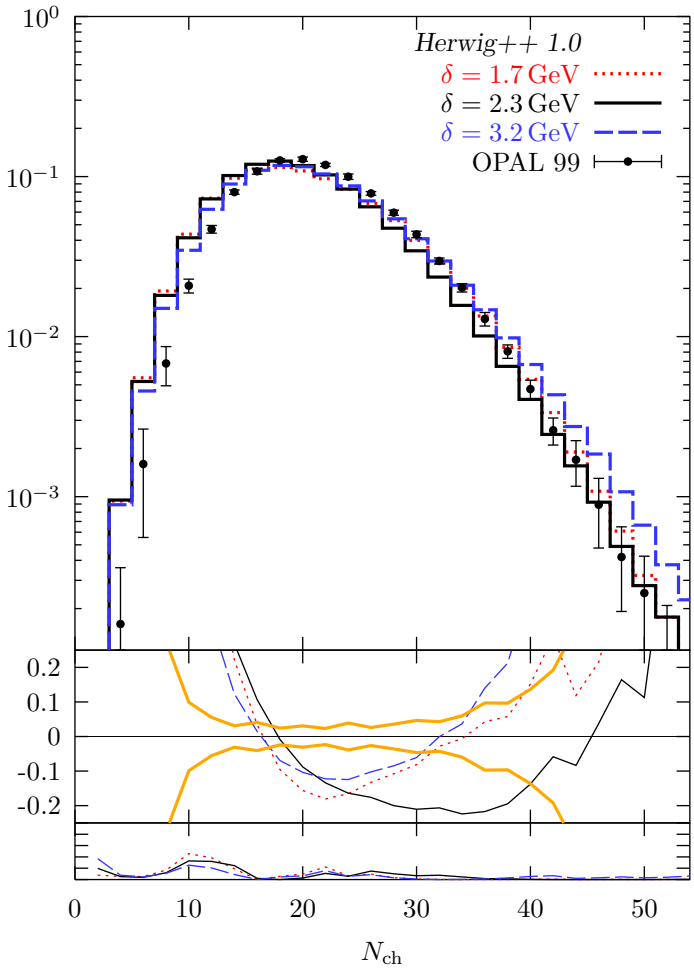


Figure 5.1: The distribution of the charged particle multiplicity.

are in better agreement with the data than those of HERWIG, with fewer results that differ from the data by more than three standard deviations (indicated by a star in the table). We can also see from table 3.3 that the difference between the models is quantified by their χ^2.

A very well measured property, and therefore important to get accurate, is the distribution of charged particle multiplicity. Figure 5.1 shows the results of Herwig++ compared to OPAL data [75] and is found to be in fairly good agreement. There is an excess of lower multiplicity events, however. It is also shown that varying δ doesn't greatly alter this distribution, another confirmation that the interface between the parton shower and the hadronization is consistent.

5.2.3 Jet multiplicity

This measurement is the multiplicity of (mini-)jets in e^+e^--collisions for different values of the jet resolution y_{cut}. We use the Durham– or $k_\perp$–clustering scheme [76] throughout this chapter for jet observables. To be specific, for a given final state the jet measure

$$y_{ij} = \frac{2\min(E_i^2, E_j^2)}{Q^2}(1 - \cos\theta_{ij}) \tag{5.2}$$

is calculated for every particle pair (i, j). The particles with minimal 'distance' are clustered such that the momentum of the clustered pseudo-particle is the sum of the four-momenta of the constituents. The jet multiplicity is then the number of pseudo-particles remaining when all $y_{ij} > y_{\text{cut}}$. This inclusive observable has been predicted and measured at LEP energies and will test the dynamics of the parton shower as well as the interface between parton shower and hadronization. We use the `KtJet`-package [77] that implements the above jet-finding algorithm in C++ and have written a simple wrapper around it in order to use it with the particle record of Herwig++.

Figure 5.2 shows the average number of jets $\langle n_{\text{jets}}\rangle$ at the Z^0–pole, as a function of the Durham jet resolution, y_{cut}, for various values of the cutoff parameter δ. At the parton level (top left) the jet multiplicity varies substantially toward smaller values of y_{cut}, saturating at the number of partons that are present in a single event. The order of magnitude of the visible saturation scales is characterized for each flavour by different cutoff values Q_g as $y_{\text{sat}} = Q_g^2/Q^2$. For example, at $Q = 91.2$ GeV and $\delta = 2.3$ GeV, the saturation scale for light quarks is of the order 10^{-4} while for b-quarks it is of the order 10^{-5}.

During hadronization, low parton multiplicities lead to large mass clusters which, as described before, tend to decay into low mass clusters below the cutoff mass, $\texttt{Cl}_{\texttt{max}}$. This has been fixed to its default value for the results given in this chapter. Figure 5.2

(top right) shows that the hadronization compensates for lower partonic multiplicities, giving a result which is insensitive to δ at the hadron level. This means that we have a smooth interface between the perturbative and non-perturbative dynamics of the lower end of the parton shower and the cluster hadronization model. On the hadron level we describe LEP data from OPAL [78] well.

In order to test the sensitivity of Herwig++ against the variation of the centre-of-mass energy we calculate the jet multiplicities at PETRA and LEP II energies as well as LEP I energies (5.2, bottom). The comparison to JADE [79] and OPAL [78] data shows a good agreement. For the generation of all the Monte Carlo data we applied the same cutoffs on the energy of the partonic subsystem as was done in the experiments.

The other curves in figure 5.2 show the prediction for the jet multiplicity [80] from the resummation of leading logarithms. Note that the parameter Λ_{QCD} in the resummed calculation is not $\Lambda_{\overline{\mathrm{MS}}}$. For a value of $\Lambda_{\mathrm{QCD}} = 500$ MeV we can see that there is good agreement with the data and the Herwig++ results throughout the perturbative region, $y_{\mathrm{cut}} > 10^{-4}$.

5.2.4 Jet fractions and Y_n

These measures give a closer look 'into' the jets. This is done by considering the rates of jets at a given value of y_{cut} in the Durham scheme. The jet fraction is given by

$$R_n = \frac{N_{n-\mathrm{jet}}}{N_{\mathrm{evts}}}, \tag{5.3}$$

for $n = 2$ up to $n = 6$ jets. Presented here are also the distributions of Y_n, the y_{cut}-values at which an n+1–jet event is merged into an n-jet event in the Durham clustering scheme. The results are presented here for $n = 2$ up to $n = 6$, without $n = 5$. These distributions will not only probe the dynamics of the parton shower but also the hadronization model;

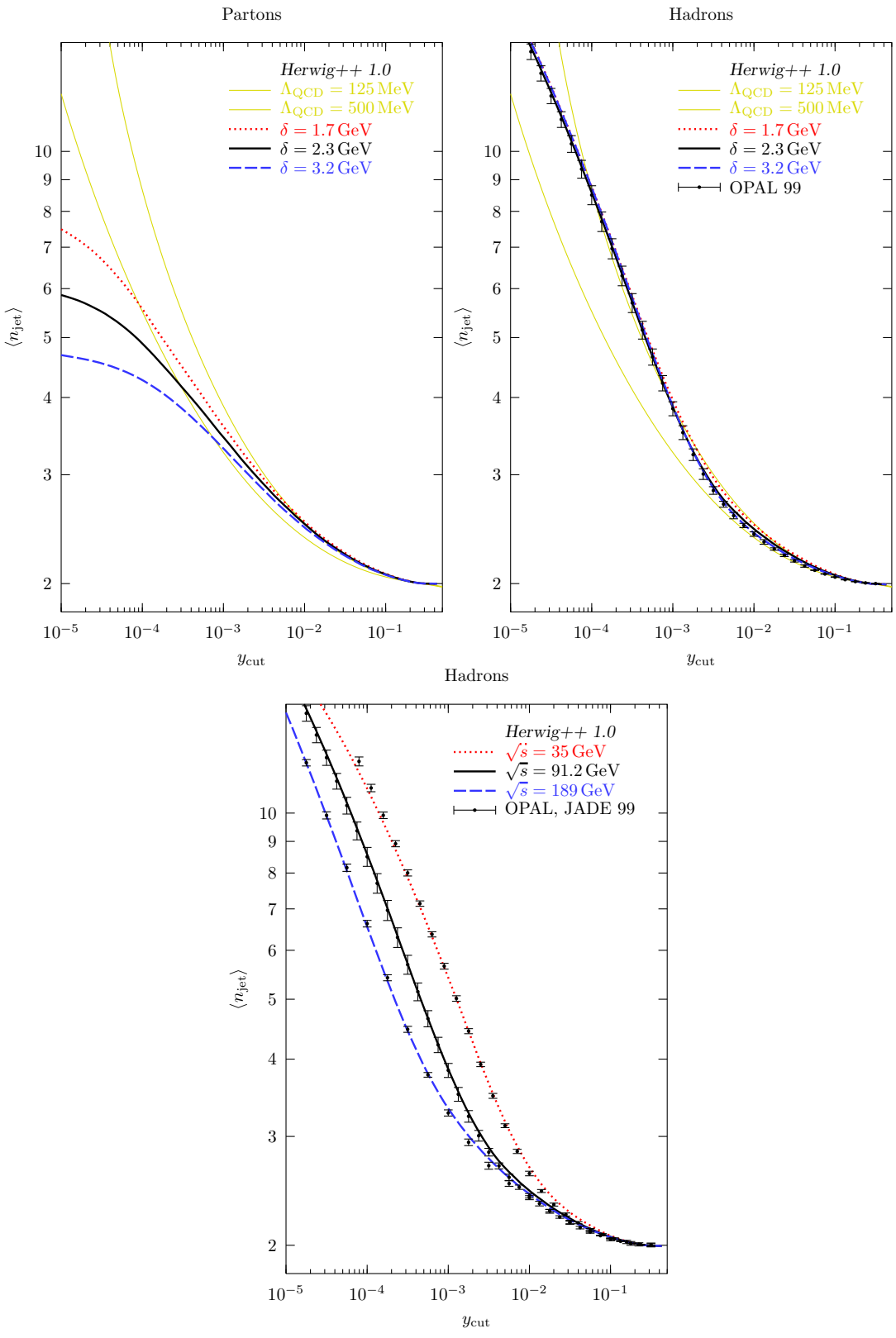


Figure 5.2: Jet multiplicities for different values of the cutoff parameter δ and different centre-of-mass energies.

at the lowest values of $y_{\rm cut} \sim (\tilde{q}_C/Q)^2$ the dynamics is dominated by the hadronization.

Figure 5.3 compares the results from Herwig++ with LEP data from [78] and shows good agreement. On the hadron level these predictions are not very sensitive to the cutoff parameter δ. The results of the R_5 are not shown here but show similarly good agreement. The R_6 plot contains all jets $n = 6$ and greater.

The Durham Y_n distributions are given in figure 5.4. These are histograms of the $y_{\rm cut}$ values at which the $n+1$–jet event in the Durham jet clustering scheme is merged into an n–jet event. This resolved more of the internal structure of the jets than the n–jet rates alone. Overall, the agreement between the model and the data is good. There is a tendency to exceed the data at low Y_n. This is a problem that was also present in HERWIG.

5.2.5 Event shapes

Event shape distributions have been measured to very high accuracy at LEP and aim at resolving the properties of the parton shower quite thoroughly. All of the event shapes given here, except S, P and A, defined below, are 'infrared safe'. This means that they can be computed in perturbation theory. In order to test the dynamics of the parton shower in Herwig++ in more detail we consider a set of commonly used event shape variables. Not only the collinear region of the parton shower is probed in greater detail but also the regions of phase space which are vetoed as matrix element corrections. We compare all results to DELPHI data [81].

The thrust is a well studied property. The definition of the thrust is given by

$$T = \max_{\boldsymbol{n}} \frac{\sum_\alpha |\boldsymbol{p}_\alpha \cdot \boldsymbol{n}|}{\sum_\alpha |\boldsymbol{p}_\alpha|}. \tag{5.4}$$

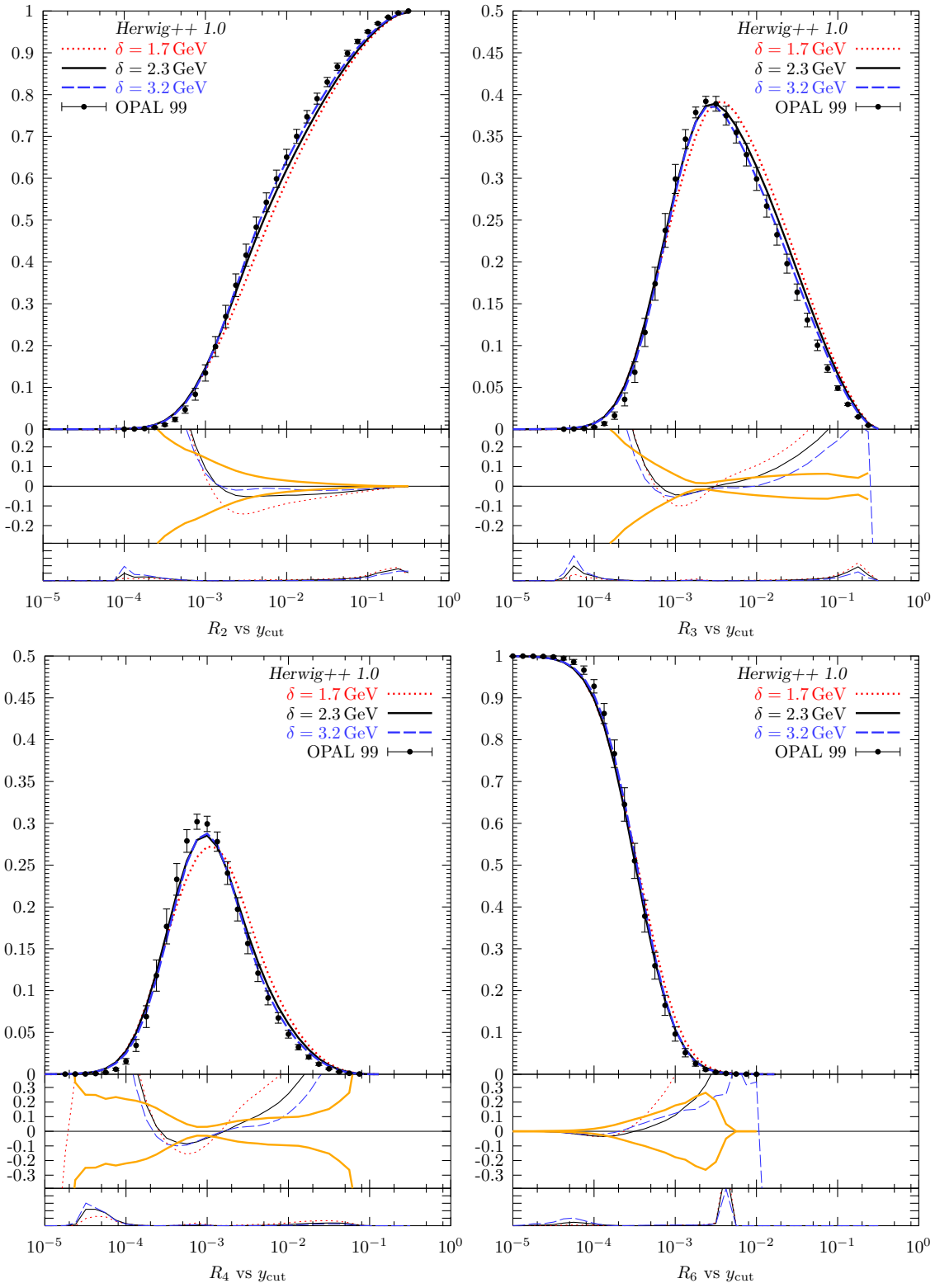


Figure 5.3: Jet rates in the Durham algorithm for different values of the cutoff δ.

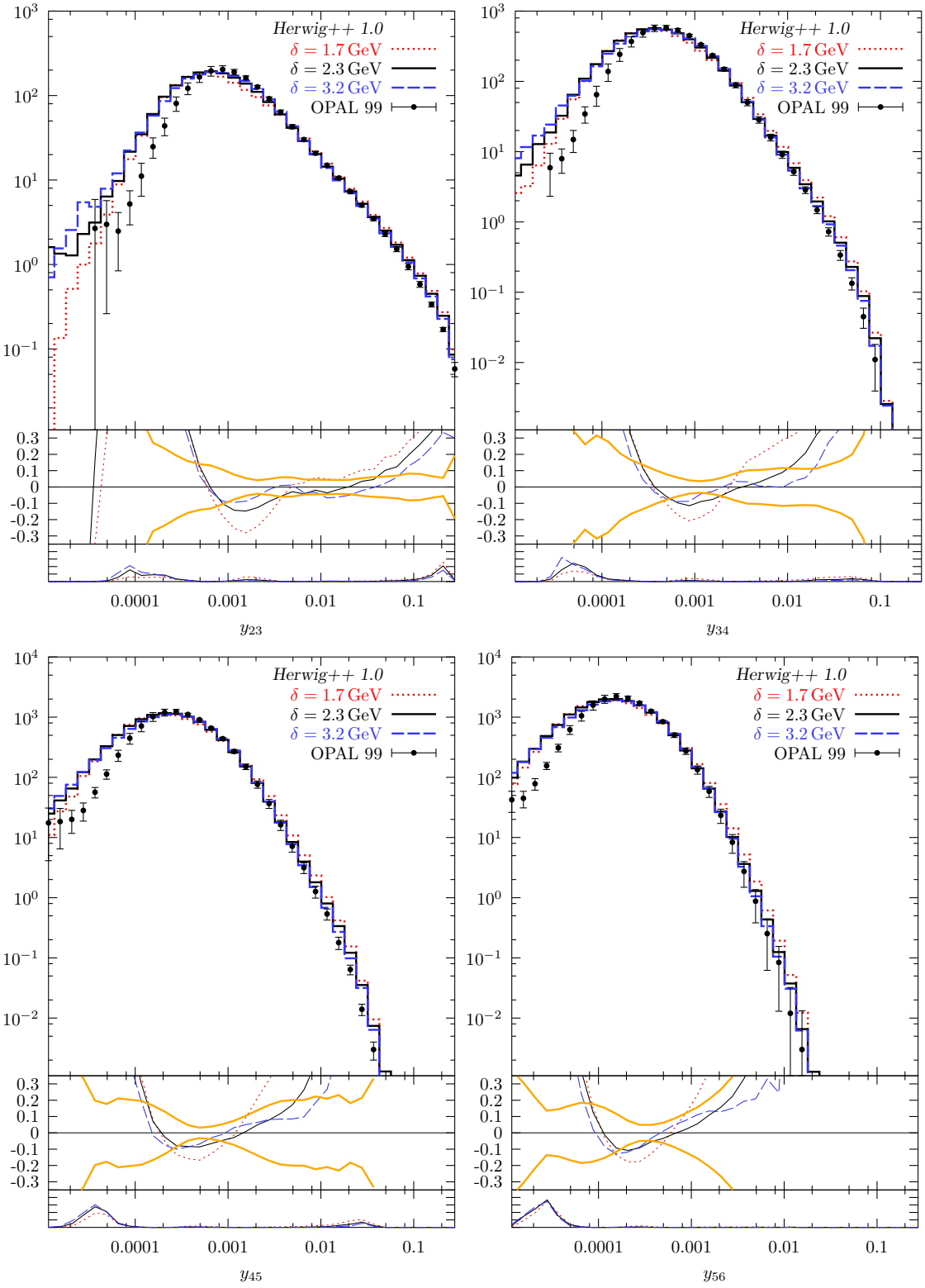


Figure 5.4: Durham Y_n distributions for different values of the cutoff δ.

Finding the vector $\boldsymbol{n}$ is a computational intensive task. This can instead be reduced using physical arguments to a simpler procedure. We start by separating the sum into two different parts: those where $\boldsymbol{p}_\alpha \cdot \boldsymbol{n} > 0$ and those where $\boldsymbol{p}_\alpha \cdot \boldsymbol{n} < 0$. If we first take out the normalization factor

$$N = \sum_\alpha |\boldsymbol{p}_\alpha|, \tag{5.5}$$

we get

$$T = \frac{1}{N} \max_{\boldsymbol{n}} \left(\sum_{\boldsymbol{p}_\alpha \cdot \boldsymbol{n} > 0} |\boldsymbol{p}_\alpha \cdot \boldsymbol{n}| + \sum_{\boldsymbol{p}_\alpha \cdot \boldsymbol{n} < 0} |\boldsymbol{p}_\alpha \cdot \boldsymbol{n}| \right). \tag{5.6}$$

The magnitudes of the dot products can be removed and this gives

$$T = \frac{1}{N} \max_{\boldsymbol{n}} \left(\sum_{\boldsymbol{p}_\alpha \cdot \boldsymbol{n} > 0} \boldsymbol{p}_\alpha \cdot \boldsymbol{n} - \sum_{\boldsymbol{p}_\alpha \cdot \boldsymbol{n} < 0} \boldsymbol{p}_\alpha \cdot \boldsymbol{n} \right). \tag{5.7}$$

Since the $\boldsymbol{n}$ is independent of the sum it can be taken outside the summation yielding

$$T = \frac{1}{N} \max_{\boldsymbol{n}} \left(\boldsymbol{P}_+(\boldsymbol{n}) - \boldsymbol{P}_-(\boldsymbol{n}) \right) \cdot \boldsymbol{n}, \tag{5.8}$$

where $\boldsymbol{P}_+(\boldsymbol{n})$ is the sum of all momenta in the same hemisphere as $\boldsymbol{n}$ and $\boldsymbol{P}_-(\boldsymbol{n})$ is the sum of all momenta in the other hemisphere. Momentum conservation says (in the c.m. frame) that $\boldsymbol{P}_+ = -\boldsymbol{P}_-$ so the thrust is given by

$$T = \frac{2}{N} \max_{\boldsymbol{n}} \boldsymbol{P}_+(\boldsymbol{n}) \cdot \boldsymbol{n}. \tag{5.9}$$

From this it is obvious that the maximum value for a given $\boldsymbol{n}$ is when the vector lies parallel to $\boldsymbol{P}_+(\boldsymbol{n})$. Previously, to find the vector $\boldsymbol{n}$ that maximizes (5.4) we would have to consider all possible combinations of momenta, which has complexity of order $N!$. After deriving (5.9) this has been reduced to considering all sets of two vectors which define a plane. This plane divides the space into two hemispheres from which the thrust can be computed by simply summing all the momenta in one of the hemispheres

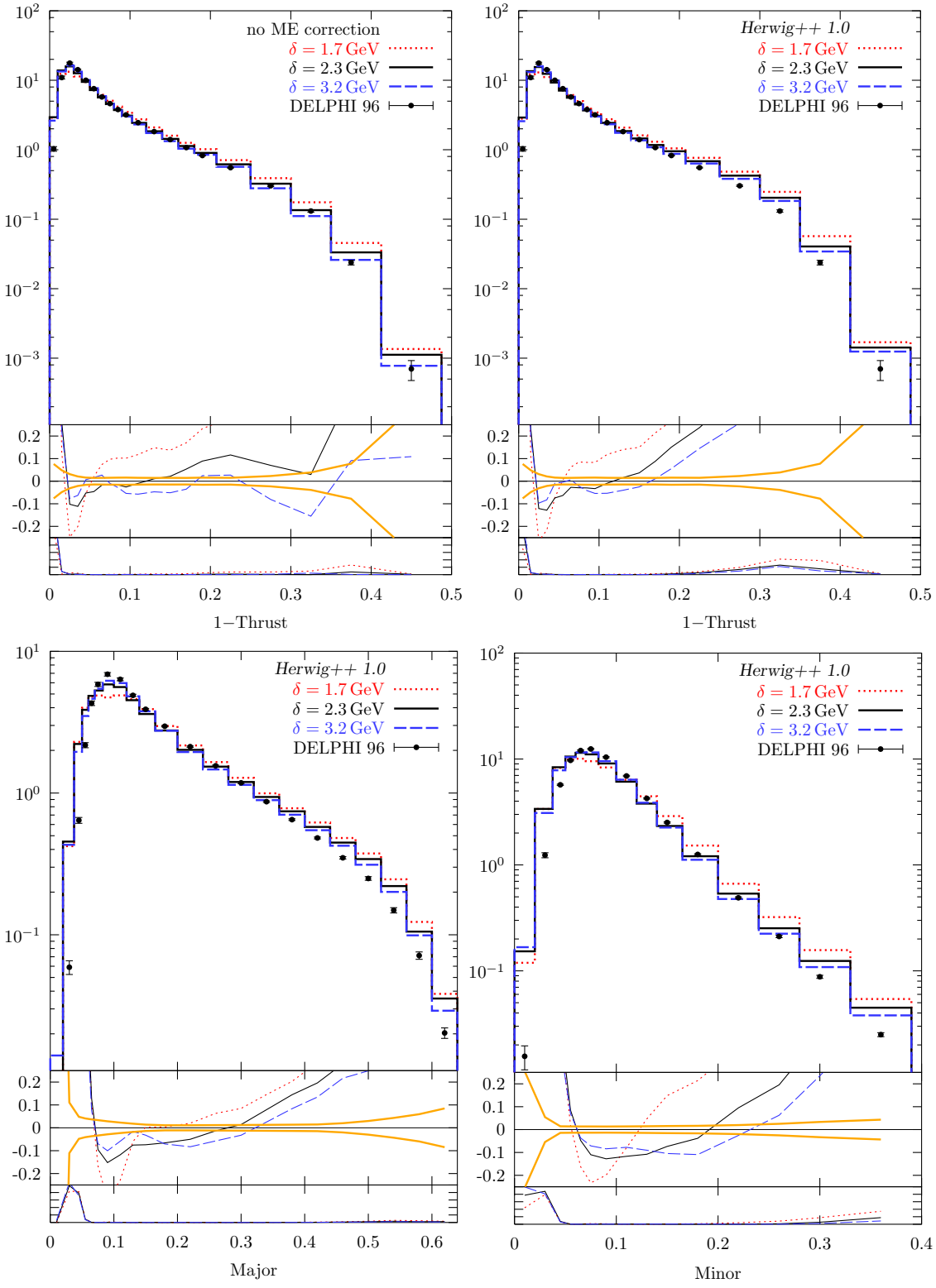


Figure 5.5: Thrust without (top left) and with (top right) matrix element corrections switched on, thrust major and thrust minor (bottom).

and taking the magnitude. Since the two vectors chosen to define the plane do not unambiguously lie in either hemisphere, each possibility must be considered. This has reduced the problem to simply iterating over all sets of two momenta and recording which set produces the maximum. This is now only order N^2.

There are two other measures of the data related to the thrust. One is called the thrust major and the other the thrust minor. These are defined as

$$M = \max_{\boldsymbol{n}\perp\boldsymbol{n}_T} \frac{\sum_\alpha |\boldsymbol{p}_\alpha \cdot \boldsymbol{n}|}{\sum_\alpha |\boldsymbol{p}_\alpha|}, \tag{5.10}$$

$$m = \frac{\sum_\alpha |\boldsymbol{p}_\alpha \cdot \boldsymbol{n}_m|}{\sum_\alpha |\boldsymbol{p}_\alpha|}, \tag{5.11}$$

where M is the thrust major, m is the thrust minor and $\boldsymbol{n}_m = \boldsymbol{n}_T \times \boldsymbol{n}_M$.

The thrust is a measure used to describe how 'pencil-like' the event is. High thrust means that the event is more 2-jet like, where as lower thrust means that the event is much more planar or spherical, thus it has more than 2 jets. The thrust major is used to describe the major component of the momentum in a plane perpendicular to the thrust axis. High values of the thrust major are usually indicative of planar events. The thrust minor, therefore, is used to describe the remaining degree of freedom of the momenta. High values of the thrust minor correspond to spherical events. These are events that have at least 4 jets.

In fig. 5.5 we show the distribution of thrust and thrust-major and thrust-minor. These variables are all obtained from the equations given above. The thrust distribution is shown with and without matrix element corrections switched on. The prediction without matrix element corrections is very much better than that of HERWIG, owing to the improved shower algorithm. It is interesting that the matrix element corrections seem to generate almost too much transverse structure, leading to event shapes that are less two-jet-like. On the other hand, there is also a slight excess of events close to the

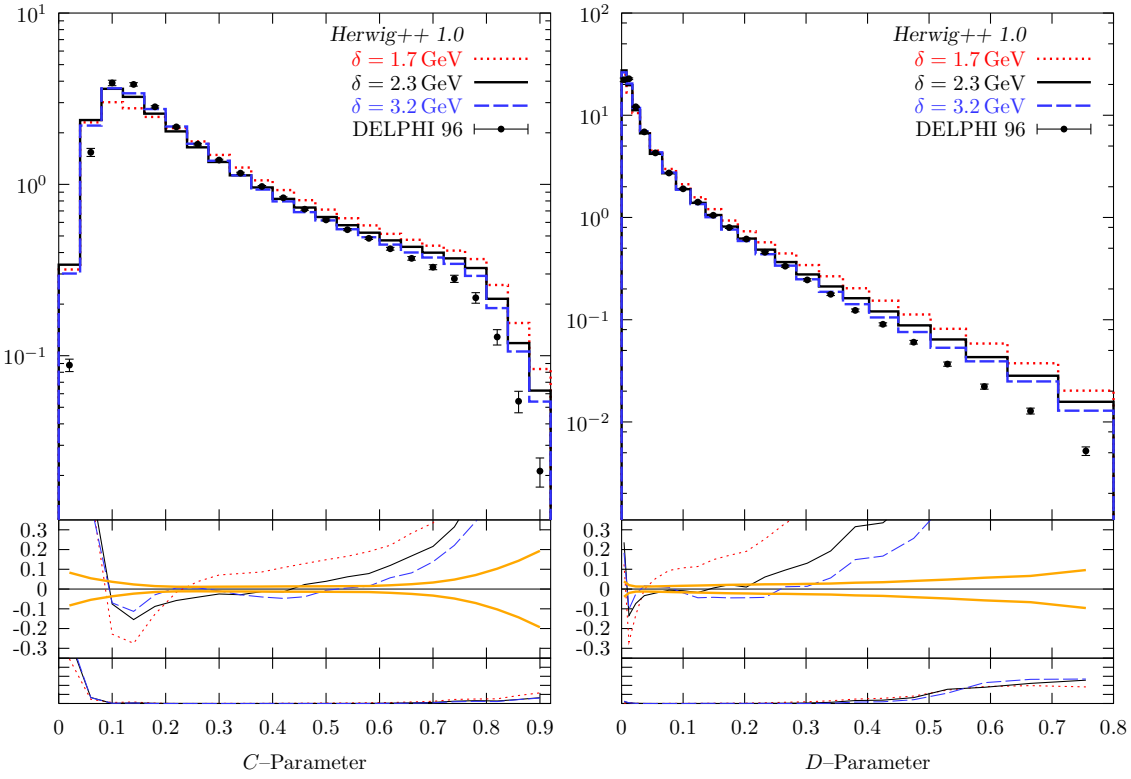


Figure 5.6: C parameter and D parameter distribution.

two-jet limit.

Though the thrust describe 2-jet like events quite well, multi-jet events are less understood in terms of these measurements. The C and D parameters are used to describe three- and four-jet-like events. These are given by combinations of the eigenvalues of the linear momentum tensor

$$L_{ij} = \frac{\sum_\alpha (\boldsymbol{p}_\alpha)_i (\boldsymbol{p}_\alpha)_j / |\boldsymbol{p}_\alpha|}{\sum_\alpha |\boldsymbol{p}_\alpha|}. \tag{5.12}$$

The definition of C and D is then

$$C = 3(\lambda_1\lambda_2 + \lambda_2\lambda_3 + \lambda_3\lambda_1), \tag{5.13}$$

$$D = 27\lambda_1\lambda_2\lambda_3, \tag{5.14}$$

where λ_i are the eigenvalues of L_{ij} and

$$\lambda_1 + \lambda_2 + \lambda_3 = 1. \tag{5.15}$$

Both of these parameters have a coefficient defined so that the range of the parameter is $[0, 1]$.

It is remarkable how well distributions like C and D parameter (fig. 5.6) which are sensitive to three- and four-jet-like events are described by our model even though we are limited to three jet matrix elements plus showers. Here again we have in fact a small excess at high values.

We show also in fig. 5.7 the distributions which are obtained from a quadratic momentum tensor

$$Q_{ij} = \frac{\sum_\alpha (\boldsymbol{p}_\alpha)_i (\boldsymbol{p}_\alpha)_j}{\sum_\alpha \boldsymbol{p}_\alpha^2}. \tag{5.16}$$

The three measures that arise from this are sphericity (S), planarity (P) and aplanarity (A). These are given by

$$S = \frac{3}{2}(\lambda_2 + \lambda_3), \tag{5.17}$$

$$P = \lambda_2 - \lambda_3, \tag{5.18}$$

$$A = \frac{3}{2}\lambda_3, \tag{5.19}$$

where λ_i is the ith eigenvalue and they obey the relations

$$\lambda_1 > \lambda_2 > \lambda_3, \quad \lambda_1 + \lambda_2 + \lambda_3 = 1. \tag{5.20}$$

These distributions put more emphasis on high momenta. As the names imply these distributions indicate the events that are spherical, planar or aplanar. The sphericity axis is just $\boldsymbol{n}_s = \boldsymbol{v}_1$ where $\boldsymbol{v}_i$ is the eigenvector of the ith eigenvalue. This is made

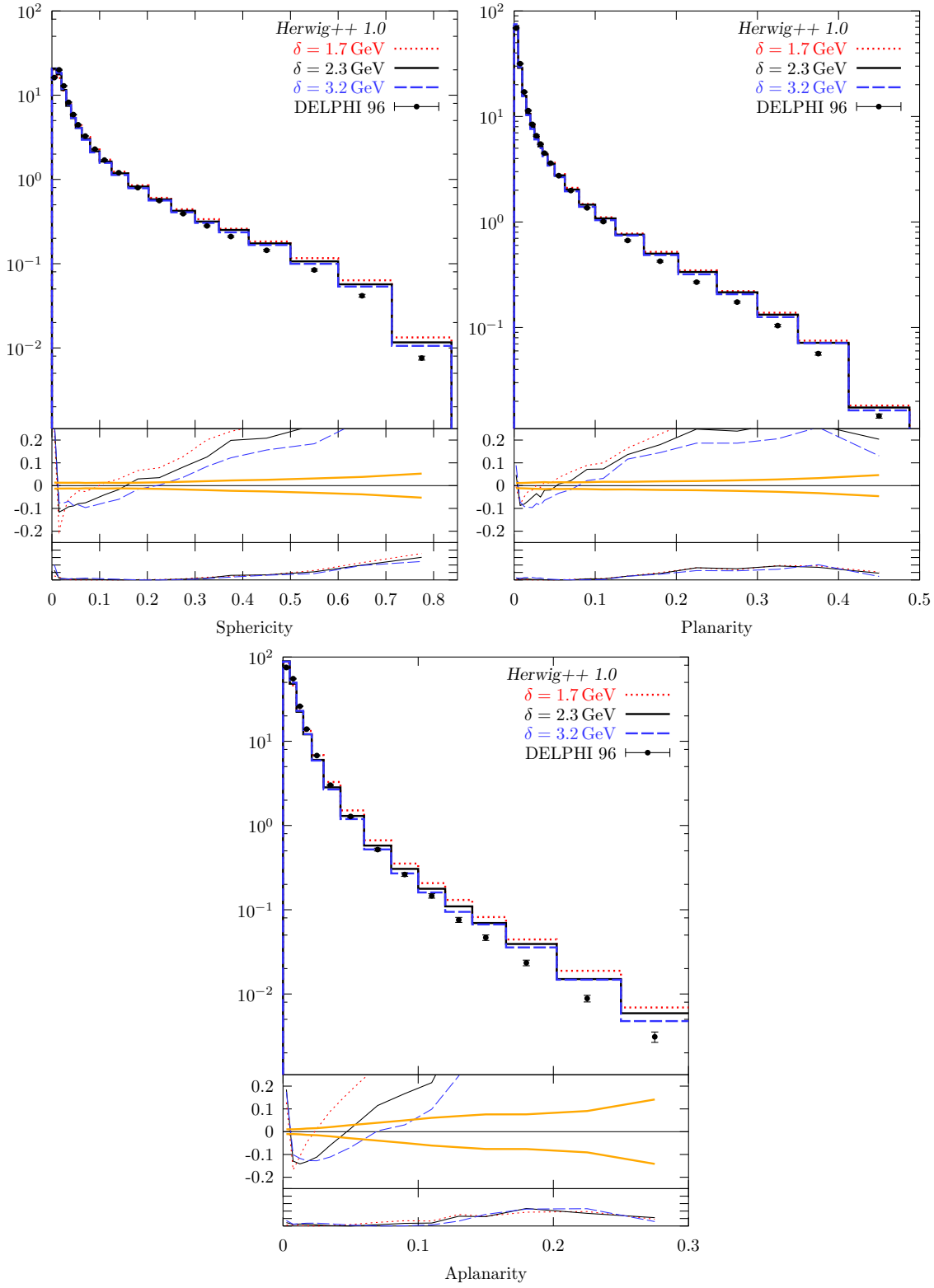


Figure 5.7: Sphericity, planarity, and aplanarity parameter distribution.

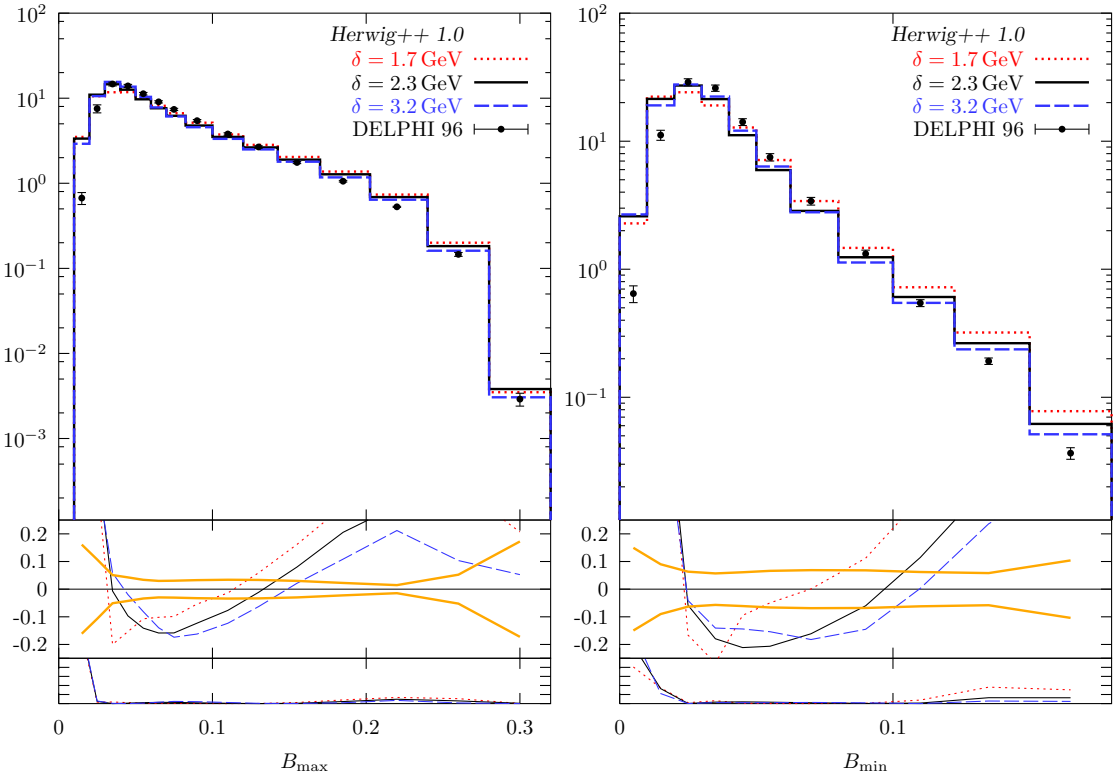


Figure 5.8: The wide and narrow jet broadening measures $B_{\max}$ and $B_{\min}$.

obvious by seeing that sphericity is really defined as $\frac{3}{2}(1-\lambda_1)$. Therefore, events which have high sphericity have momentum which tend to be along the sphericity axis, $\boldsymbol{n}_s$. As was the case for the thrust-related distributions, we tend to have slightly wide events.

In addition we consider the jet broadening measures $B_{\max}$ and $B_{\min}$ and the hemisphere jet masses (fig. 5.8 and fig. 5.9). The jet broadening measures are defined as [82, 83]

$$B_{\max} = \max_{i=1,2} \frac{\sum_{\boldsymbol{p}_k \in H_i} |\boldsymbol{p}_k \times \boldsymbol{n}_T|}{2\sum_k |\boldsymbol{p}_k|}, \tag{5.21}$$

where $\boldsymbol{n}_T$ is the thrust axis and H_i indicates one of the two hemispheres defined by the plane normal to $\boldsymbol{n}_T$. If $\boldsymbol{p}_k \cdot \boldsymbol{n}_T > 0$ then $\boldsymbol{p}_k$ is in hemisphere, H_1, otherwise it is in hemisphere H_2. $B_{\min}$ is then

$$B_{\min} = \min_{i=1,2} \frac{\sum_{\boldsymbol{p}_k \in H_i} |\boldsymbol{p}_k \times \boldsymbol{n}_T|}{2\sum_k |\boldsymbol{p}_k|}. \tag{5.22}$$

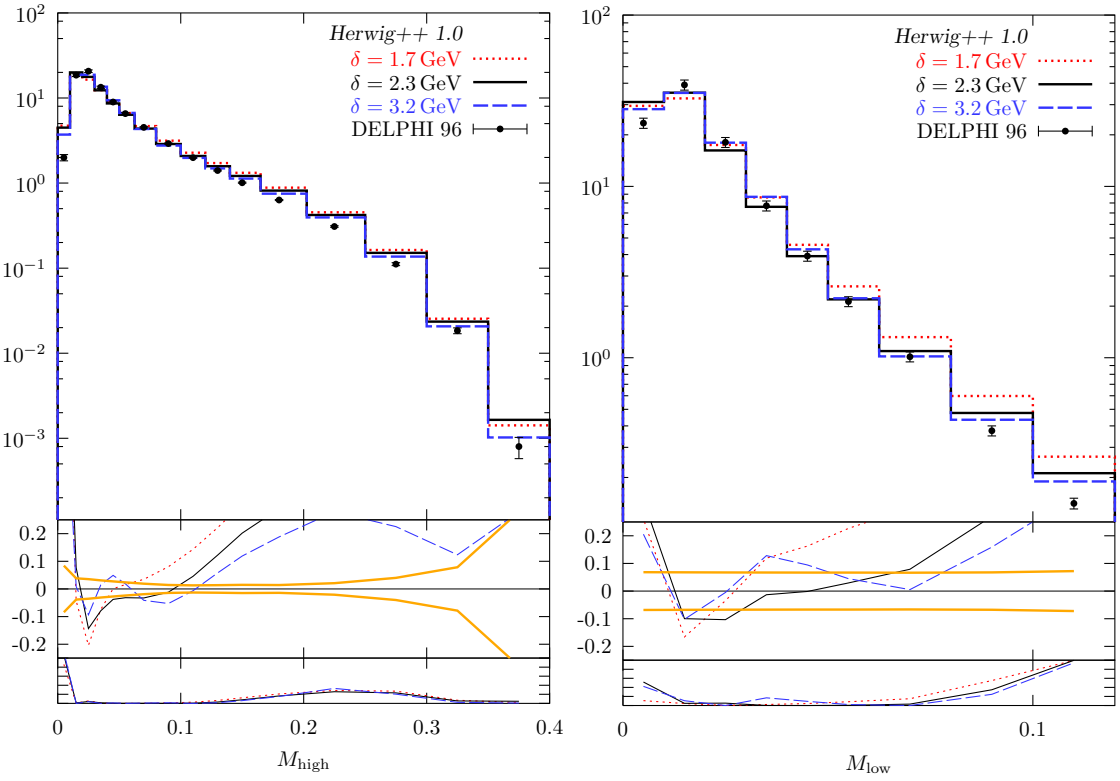


Figure 5.9: The high and low hemisphere masses.

$B_{\max}, B_{\min}$ measure the scalar momentum transverse to the thrust axis for the wider and narrower jet hemispheres respectively. We can see from figure 5.8 that there is good agreement between the model and the data. We have also looked at two other jet broadening measures, $B_{\text{diff}} = B_{\max} - B_{\min}$ and $B_{\text{sum}} = B_{\max} + B_{\min}$. These are not shown here as they contain the same information as $B_{\max}$ and $B_{\min}$.

The last jet measure we show here is that of the hemisphere masses. These also use the same definitions of the two hemispheres with respect to the thrust axis as the jet broadening measures, but these measure the total momentum squared within a hemisphere. The high hemisphere mass is

$$M_{\text{high}} = \frac{1}{s} \max_{i=1,2} \left(\sum_{\boldsymbol{p}_k \in H_i} p_k \right)^2 , \tag{5.23}$$

and the lower hemisphere mass is

$$M_{\text{low}} = \frac{1}{s} \min_{i=1,2} \left(\sum_{\boldsymbol{p}_k \in H_i} p_k \right)^2 , \tag{5.24}$$

where s is the c.m. energy squared. In both cases we can see from figure 5.9 that the agreement between model and data is good.

5.2.6 Four jet angles

We show the four-jet angles in fig. 5.10. They are considered only for events where we have a four-jet event at $y_{\text{cut}} = 0.008$. Each of the different angles measures a property of the four-jet configuration.

If we denote the cosine of the angle between two vectors, $\boldsymbol{a}$ and $\boldsymbol{b}$ as $C(\boldsymbol{a}, \boldsymbol{b})$, we can define the four different jet angles as

$$\begin{aligned}
\cos\chi_{\text{BZ}} &= C\left((\boldsymbol{p}_1 \times \boldsymbol{p}_2), (\boldsymbol{p}_3 \times \boldsymbol{p}_4)\right), && (5.25)\\
\cos\Phi_{\text{KSW}} &= \cos\left[\frac{\alpha_1 + \alpha_2}{2}\right], && (5.26)\\
\cos\Theta_{\text{NR}} &= C\left((\boldsymbol{p}_1 - \boldsymbol{p}_2), (\boldsymbol{p}_3 - \boldsymbol{p}_4)\right), && (5.27)\\
\cos\alpha_{34} &= C\left(\boldsymbol{p}_3, \boldsymbol{p}_4\right), && (5.28)
\end{aligned}$$

where

$$\begin{aligned}
\cos\alpha_1 &= C\left((\boldsymbol{p}_1 \times \boldsymbol{p}_4), (\boldsymbol{p}_2 \times \boldsymbol{p}_3)\right), && (5.29)\\
\cos\alpha_2 &= C\left((\boldsymbol{p}_1 \times \boldsymbol{p}_3), (\boldsymbol{p}_2 \times \boldsymbol{p}_4)\right). && (5.30)
\end{aligned}$$

$\boldsymbol{p}_1$ is the three-momentum of the hardest jet and $\boldsymbol{p}_4$ is the three-momentum of the softest jet.

Despite the fact that we do not have any matching to higher order matrix elements, as was proposed in [27] and implemented in [84], the agreement between model and data [85] is remarkably good. We expected the implementation of hard and soft matrix element corrections in Herwig++ to improve the description of these observables but we did not find very significant differences with or without the application of matrix element corrections.

5.2.7 Single particle distributions

Using the thrust axis defined in (5.4), one can define the transverse momentum of a particle with respect to this axis. We can then define a plane given by the thrust axis and the thrust major axis, $\boldsymbol{n}_M$ as the event plane. Therefore, the plane given by the thrust and the thrust minor, $\boldsymbol{n}_m$, is considered out of the event plane. In fig. 5.11 we have single charged particle distributions. The distributions are of the transverse momentum within the event plane, $p^T_{\perp,\text{in}}$, and the transverse momentum out of the event plane, $p^T_{\perp,\text{out}}$. The momenta in the event plane are shown with and without matrix element corrections. In contrast to the thrust distribution we find that the matrix element corrections actually improve the distribution. Furthermore, $p^T_{\perp,\text{out}}$ and the rapidity along the thrust axis, y_T, are rather well described. A similar technique can be used to define the transverse momentum with respect to the sphericity axis $\boldsymbol{n}_s$. We do not show these but they have similar features to the distributions with respect to the thrust axis.

As we saw in chapter 3, we can consider the distribution of scaled momentum $x_p = 2|\boldsymbol{p}|/Q$ of charged particles. The results in chapter 3 were given to show that the hadronization technique could describe these distributions quite well. In fig. 5.12 these distributions are given to show their dependence on the shower cutoff, δ. In addition to the full distribution we also consider the results from light (uds), c and b events. [a] In

[a] The flavour of the quark-antiquark produced in the initial hard process

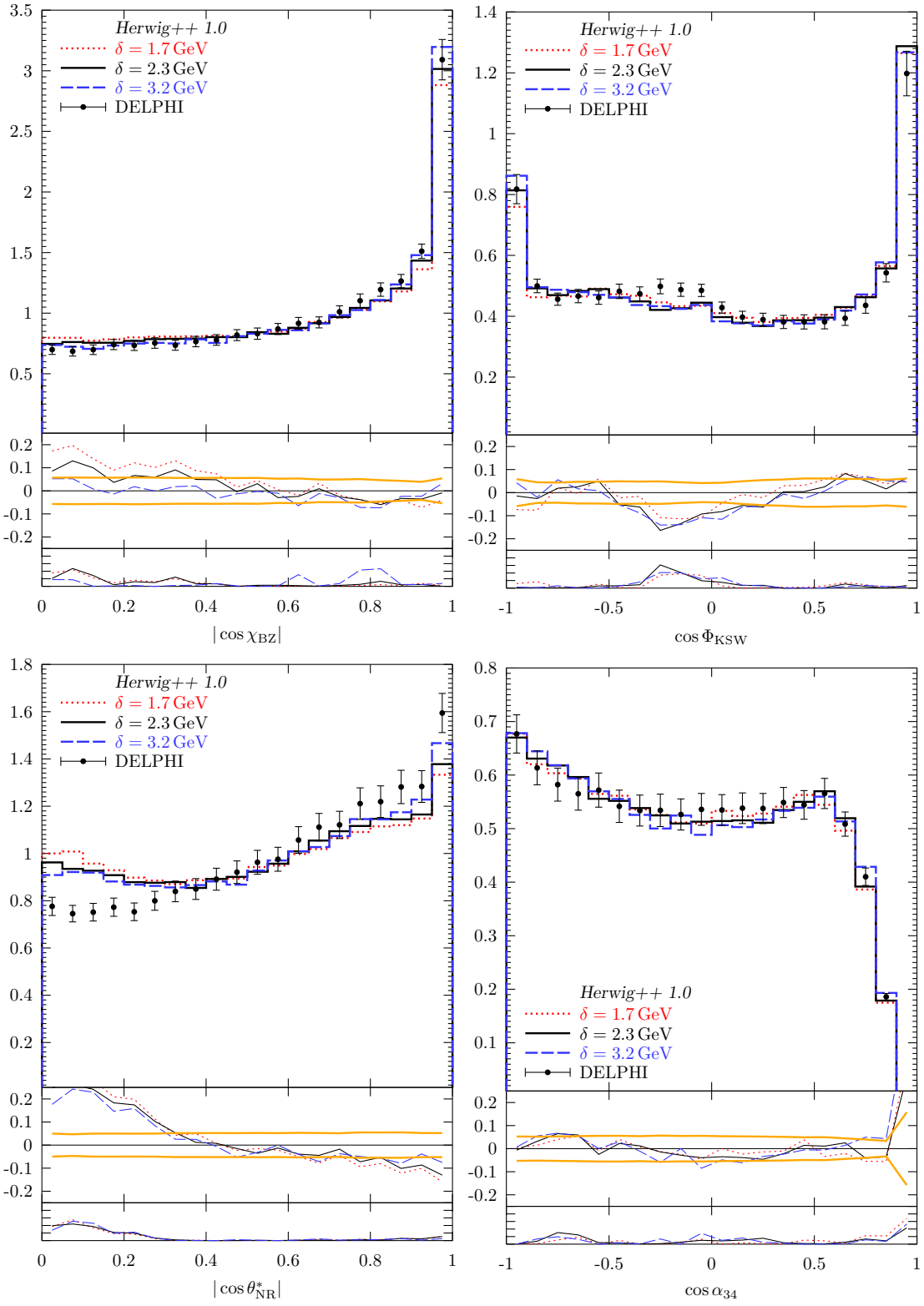


Figure 5.10: Four jet angle distributions. The points are from preliminary DELPHI data.

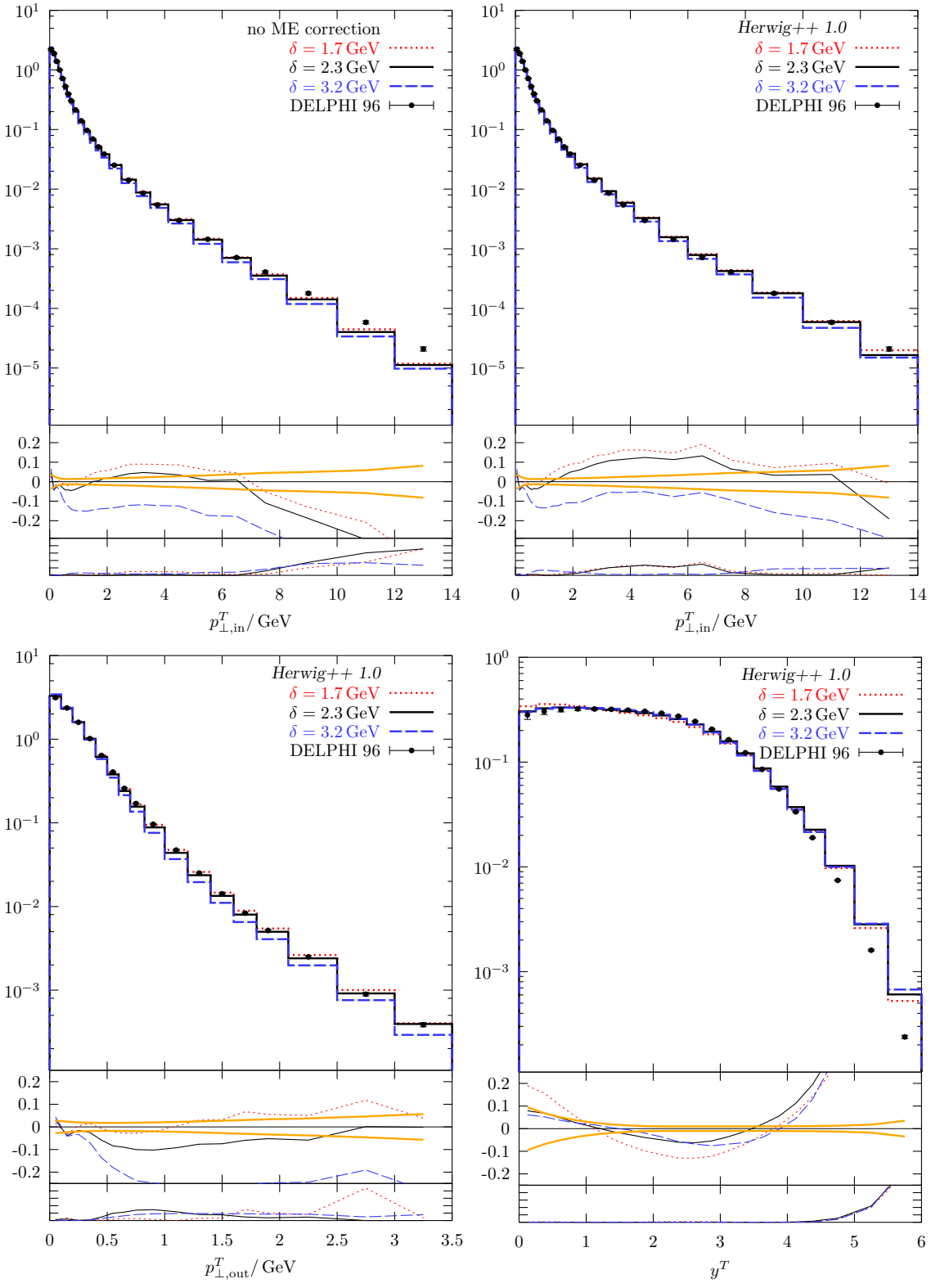


Figure 5.11: Momentum distributions of charged particles with respect to the thrust axis, $p^T_{\perp,\mathrm{in}}$ (with and without matrix element corrections), $p^T_{\perp,\mathrm{out}}$ and y^T.

all cases we compare with data from SLD [86]. The charged particle distribution is well described in all four cases, in fact somewhat better for heavy primary quarks.

5.2.8 Identified hadron spectra

As in the case of all charged particles we can compare identified particle spectra from events of different flavour to SLD data [86]. Data for $\pi^{\pm}$ (not shown, being almost equivalent to all charged particles), $K^{\pm}$ and $(p, \bar{p})$ is available. In fig. 5.13 we see the data for $(p, \bar{p})$ spectra from events of different flavour. For large values of x_p we clearly overshoot the data in light flavoured events, producing the 'bump'. This is somewhat compensated by the heavy quark events which in turn seem to prefer lower values of x_p. The origin of the 'bump' is not well understood. We believe that this feature is related to the hadronization, being similar to but smaller than that seen in HERWIG, but this has not been shown conclusively.

Fig. 5.14 shows distributions for $K^{\pm}$ and $\Lambda, \bar{\Lambda}$. Both are rather better described than the proton spectra but the distribution of $\Lambda, \bar{\Lambda}$ tends to have a similar, though smaller, 'bump' in comparison to data from ALEPH [87].

5.2.9 B fragmentation function

The hadron fragmentation function is defined as the distribution of

$$x_E = \frac{2E_h}{E_{\rm cm}}, \tag{5.31}$$

where E_h is the energy of the hadron. Using this we can look at the fragmentation function of different flavours or species of hadrons. In fig. 5.15 we consider the B hadron fragmentation function in comparison to data from SLD [88]. This is the fragmentation

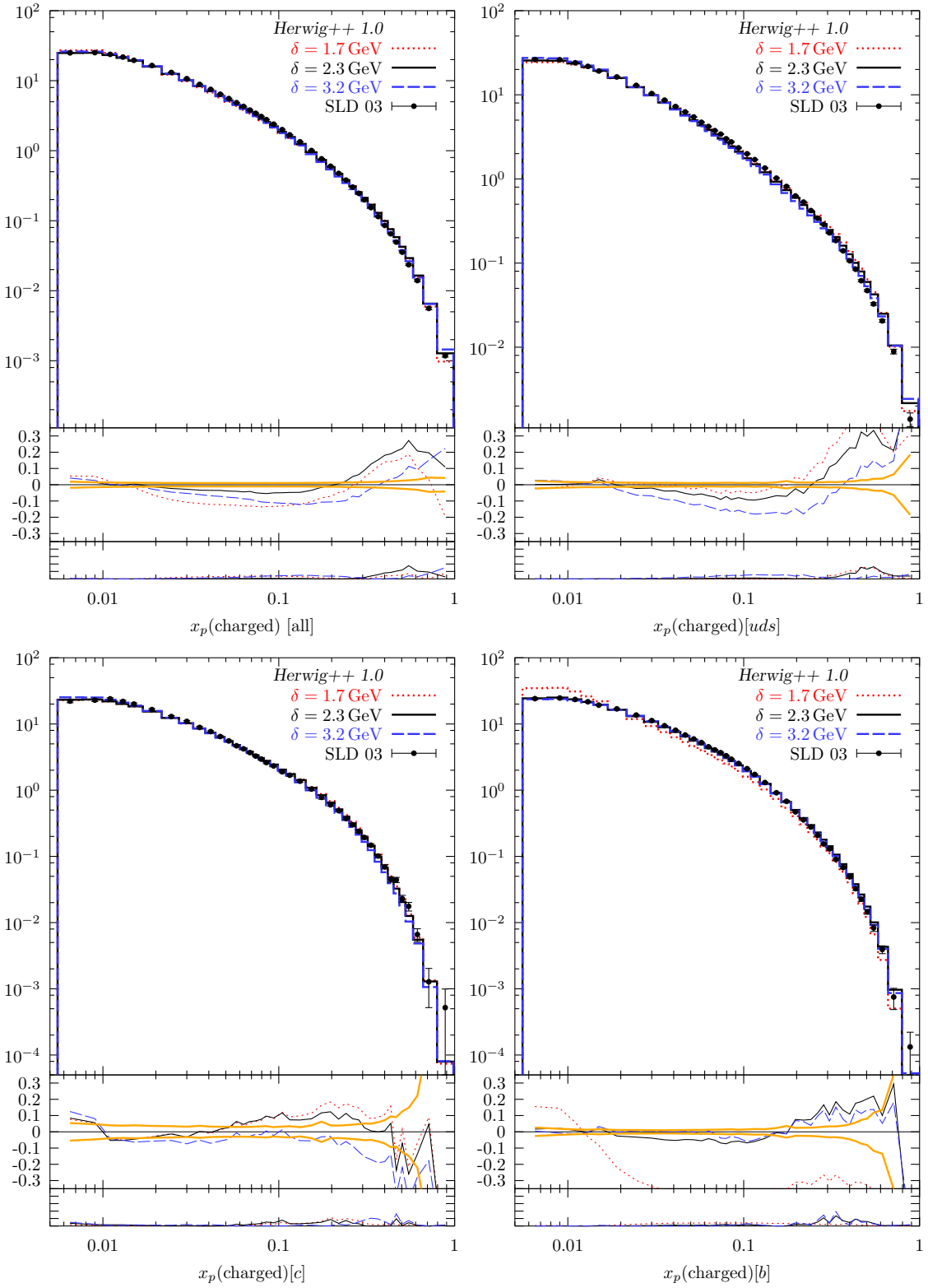


Figure 5.12: The scaled momentum distribution x_p of charged particles for all events as well as for uds, c and b events separately.

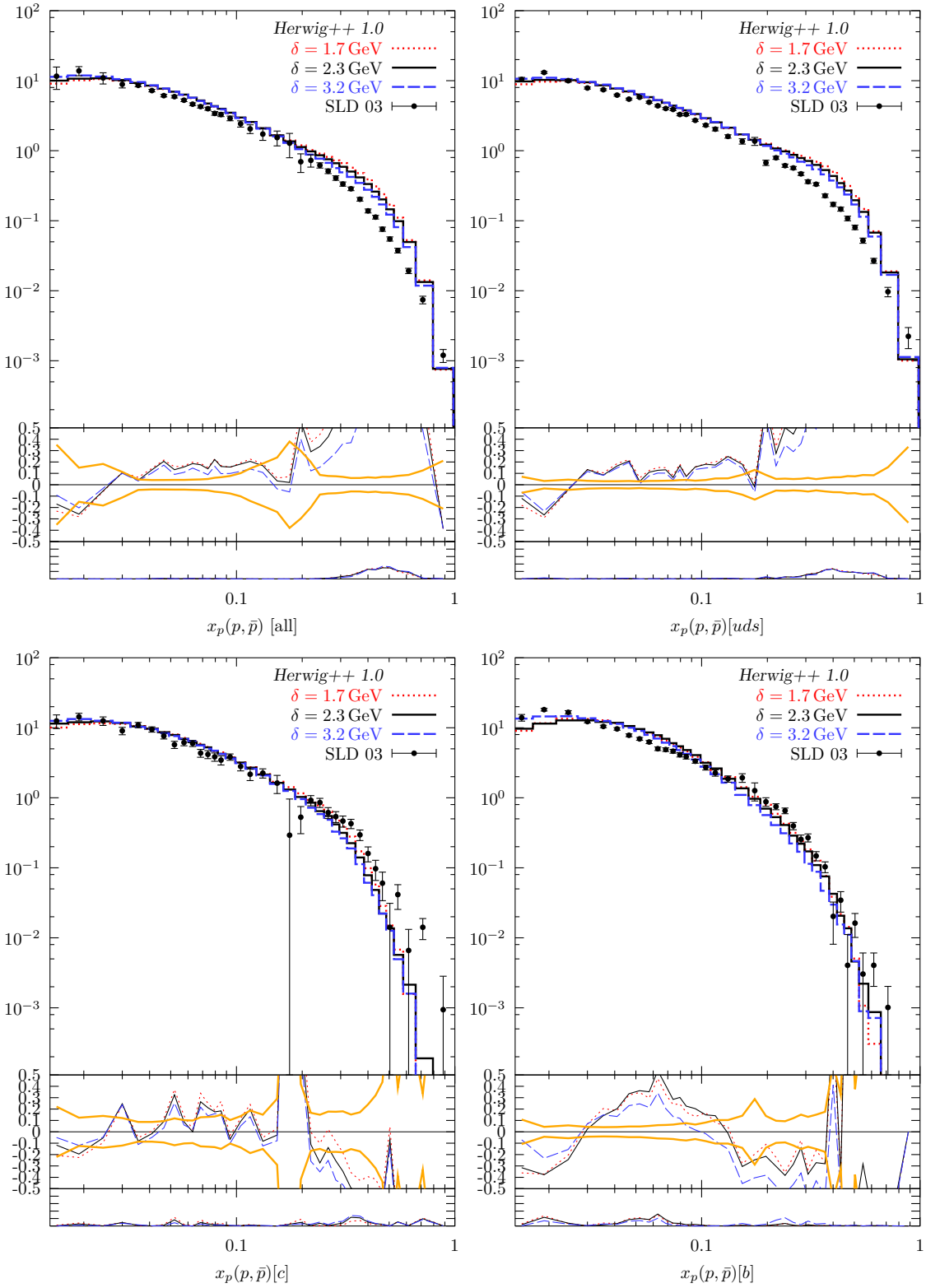


Figure 5.13: The scaled momentum distribution x_p of protons, shown separately for all events as well as for uds, c and b events.

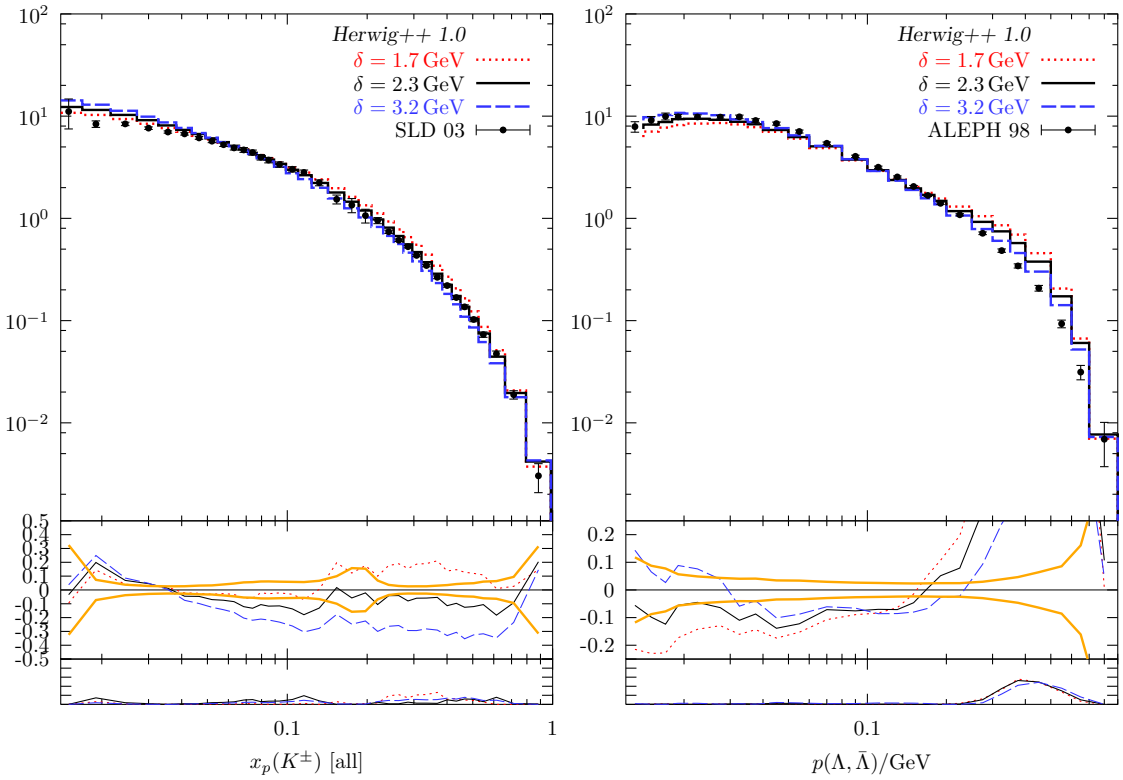


Figure 5.14: Distribution of scaled Kaon momentum and $\Lambda, \bar{\Lambda}$ momentum.

function for all weakly decaying B hadrons. We have also considered data from ALEPH [89] for comparison, though these are not shown here. We can describe the data quite well without any additional tuning of the hadronization model to this data. The parton shower formulation in terms of the new variables [36] and taking quark masses in the splitting functions into account clearly improves the description of heavy quark events.

From figure 5.15 we tend to bias the fit towards the δ of 2.3 GeV, as the improved treatment of the b quarks was the main motivation for deriving the new variables and using the massive splitting functions. HERWIG couldn't describe the data as well even with the extra flags added to the hadronization model to parameterize B hadrons differently.

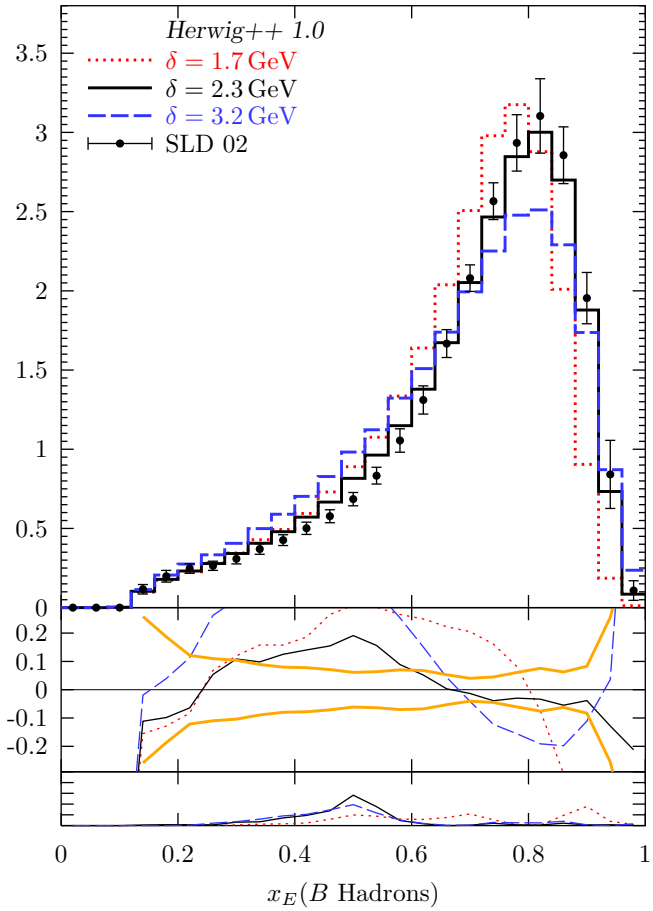


Figure 5.15: The B–hadron fragmentation function. For different values of the cutoff δ.

5.2.10 Overall results of e^+e^- annihilation

In Tab. 5.2 we show a list of χ^2 values for all observables that were studied during our analysis, including those not shown in the plots. The most sensitive parameters were the cutoff value δ and the use of (hard plus soft) matrix element corrections. The table shows three values of δ: our preferred value of $\delta = 2.3\,\mathrm{GeV}$ as well as the lowest and highest values that we considered.

The results should be interpreted with care. The overall trend suggests that we should prefer a large cutoff scale. However, we have just averaged over all possible observables. Taking a closer look, we may want to weight different observables in a different way.

In more detail, the general trend is the following: event shapes, jet rates and differential jet rates prefer a low cutoff. The single particle distributions along the thrust and sphericity axes prefer a small cutoff value. The y_{nm} distributions prefer either a high or a low cutoff value. The spectra of identified particles tend to prefer the high cutoff value with some exceptions for light quark events. The B fragmentation function clearly prefers the intermediate value.

In addition, as indicated in sect. 5.2.1, we found that the measured yields of identified particles clearly prefer the value $\delta = 2.3\,\mathrm{GeV}$.

5.3 Conclusions

We have achieved a complete event generator for e^+e^- annihilation into hadrons. The main physics features, in comparison to the previous versions of HERWIG, are an improved parton shower, capable of properly describing the perturbative splitting of heavy quarks, and an improved cluster hadronization model.

We have tested our model against a wide range of data from e^+e^- colliders and are

		ME corrections off			ME corrections on		
Observable	Ref.	$\delta = 1.7\,\mathrm{GeV}$	$\delta = 2.3\,\mathrm{GeV}$	$\delta = 3.2\,\mathrm{GeV}$	$\delta = 1.7\,\mathrm{GeV}$	$\delta = 2.3\,\mathrm{GeV}$	$\delta = 3.2\,\mathrm{GeV}$
$1-T$	[81]	44.65	33.06	22.29	70.34	44.14	25.99
M	[81]	246.25	265.81	198.37	257.99	242.19	174.50
m	[81]	150.74	155.11	137.43	167.05	150.10	120.59
O	[81]	7.41	5.60	5.14	21.48	19.27	12.78
S	[81]	4.42	3.48	4.07	22.99	13.09	7.89
P	[81]	4.48	5.53	6.54	10.27	7.34	5.35
A	[81]	19.52	10.75	7.17	41.86	18.26	9.62
C	[81]	66.86	59.08	39.56	79.85	64.93	42.32
D	[81]	84.23	28.40	12.36	145.14	51.44	23.96
M_{high}	[81]	25.78	18.82	12.38	37.80	25.39	11.34
M_{low}	[81]	15.25	5.36	2.50	30.29	9.93	4.78
M_{diff}	[81]	7.28	5.18	7.25	17.62	12.26	4.42
B_{max}	[81]	54.48	50.47	38.91	59.06	49.27	33.32
B_{min}	[81]	53.25	55.83	53.18	63.65	56.69	49.91
B_{sum}	[81]	102.29	97.12	74.60	116.81	98.56	69.94
B_{diff}	[81]	8.28	5.56	4.70	17.91	13.77	5.94
$p_{\perp,\mathrm{in}}^{T}$	[81]	2.48	3.14	11.52	3.22	1.78	4.17
$p_{\perp,\mathrm{out}}^{T}$	[81]	0.25	3.25	21.65	0.71	1.69	16.11
y^{T}	[81]	34.52	59.78	66.05	32.88	49.36	55.54
$p_{\perp,\mathrm{in}}^{S}$	[81]	2.53	3.18	11.76	2.21	1.40	4.28
$p_{\perp,\mathrm{out}}^{S}$	[81]	0.37	3.78	22.64	1.01	2.02	16.67
y^{S}	[81]	9.04	17.42	24.85	7.53	13.80	21.35
D_2^D	[81]	9.37	3.65	3.76	25.14	11.81	5.17
D_3^D	[81]	25.85	6.32	2.14	46.56	15.68	5.39
D_4^D	[81]	43.90	10.56	2.69	77.40	22.61	7.16
y_{23}	Ø	8.75	6.19	5.36	12.17	8.41	6.27
y_{34}	Ø	10.20	9.49	9.07	11.42	9.31	8.64
y_{45}	Ø	15.53	14.33	11.78	17.07	14.47	11.66
y_{56}	Ø	16.02	17.62	15.13	15.28	16.56	13.82
$\langle N_{\mathrm{jets}} \rangle$	Ø	12.84	3.38	0.62	27.84	13.11	5.81
R_2	Ø	9.75	6.64	6.18	19.71	13.40	9.55
R_3	Ø	10.46	8.51	9.36	23.45	15.76	12.02
R_4	Ø	13.47	11.06	10.36	15.45	12.69	10.29
R_5	Ø	25.53	25.18	23.43	27.88	26.38	22.40
R_6	Ø	10.37	1.80	0.67	18.98	4.31	1.41
$\cos(\chi_{\mathrm{BZ}})$	[85]	2.90	1.04	0.48	2.60	1.05	0.45
$\cos(\Phi_{\mathrm{KSW}})$	[85]	2.30	1.99	2.56	1.30	1.63	1.71
$\cos(\theta^*_{\mathrm{NR}})$	[85]	7.68	4.82	2.72	8.57	5.52	3.74
$\cos(\alpha_{34})$	[85]	1.41	1.47	1.71	0.51	0.46	0.74
N_{ch}	[75]	21.86	25.68	12.90	19.88	22.55	13.07
$x_p(\mathrm{ch})[\mathrm{all}]$	[86]	5.32	5.65	3.49	4.77	4.10	3.02
$x_p(\mathrm{ch})[uds]$	[86]	15.72	8.49	6.13	12.70	6.69	5.82
$x_p(\mathrm{ch})[c]$	[86]	3.95	2.33	2.17	2.95	1.72	2.72
$x_p(\mathrm{ch})[b]$	[86]	35.05	3.23	1.79	35.80	2.46	1.15
$x_p(\pi^{\pm})[\mathrm{all}]$	[86]	8.29	9.31	6.18	7.17	7.52	5.52
$x_p(\pi^{\pm})[uds]$	[86]	28.30	15.99	10.47	23.71	13.19	9.45
$x_p(\pi^{\pm})[c]$	[86]	4.65	3.04	1.38	3.67	2.24	1.60
$x_p(\pi^{\pm})[b]$	[86]	49.13	3.13	1.56	49.37	3.69	2.05
$x_p(K^{\pm})[\mathrm{all}]$	[86]	4.99	2.01	15.38	3.67	2.84	17.41
$x_p(K^{\pm})[uds]$	[86]	6.46	16.98	36.45	6.79	19.23	38.67
$x_p(K^{\pm})[c]$	[86]	21.01	2.22	3.35	17.71	1.74	4.24
$x_p(K^{\pm})[b]$	[86]	8.56	7.07	4.34	7.63	5.74	4.97
$x_p(p,\bar{p})[\mathrm{all}]$	[86]	143.34	96.18	42.90	135.35	80.31	34.27
$x_p(p,\bar{p})[uds]$	[86]	145.35	100.90	52.78	135.61	85.09	43.31
$x_p(p,\bar{p})[c]$	[86]	2.26	2.38	2.86	2.34	2.51	2.87
$x_p(p,\bar{p})[b]$	[86]	11.26	13.54	8.12	10.98	13.31	8.34
$p(\Lambda,\Lambda)$	[87]	58.02	28.32	9.47	53.88	24.92	7.69
$x_E(B)$	[88]	8.93	0.95	8.16	9.41	1.35	9.93
$x_E(B)$	[89]	15.40	1.74	7.35	15.77	2.02	8.23
$\langle\chi^2\rangle/\mathrm{dof}$		32.75	25.54	20.93	39.22	26.72	19.84

Table 5.2: χ^2 values for all observables we studied and a relevant subset of parameters.

able to give a good general description of the data.

For many observables the description of the data has been improved with respect to HERWIG. The new parton shower has a number of remarkable features. The need for matrix element corrections has decreased. The main reason for this is the use of improved splitting functions, which give a far better approximation of the matrix elements in the region of collinear gluon emissions. We can describe observables involving light or heavy quark splitting with a unique set of parameters. The new hadronization model also improves the description of identified particle spectra and multiplicities.

The detailed analysis of our results leaves us with a *recommendation*: the set of parameters that is shown in table 5.1. This set of parameters is understood as a weighted compromise in order give a good overall description of the data we have considered so far. We did not aim at a complete tuning of the model, but rather wanted to study its ability to describe the broad features of the data, which turned out to be very successful.

Work is currently under way testing the parton shower on initial state radiation. A model for the soft underlying event in hadron–hadron collisions is also under development. The aim is to have the code tested and debugged so a complete event generator for the simulation of Tevatron and LHC events is available.

Chapter 6

Effective Potential Analysis: Effective

6.1 Supersymmetry

Though the SM presently describes physical phenomena extremely well there is reason to believe that it will eventually be insufficient to describe physics at higher scales. The scale tested to date is of the order of 100 GeV. The Planck scale, $M_P = (8\pi G_{\text{Newton}})^{-1/2} = 2.4 \times 10^{18}$ GeV, is the scale at which gravitational effects will be of the same order as the other interactions. This scale is 16 orders of magnitude higher than the currently observed phenomena. On an intuitive basis this is suggestive that there must be new phenomena occurring between these scales. This is commonly referred to as the hierarchy problem.

There is also another problem, known as the fine tuning problem. This is due to the Higgs boson. The mass of the Higgs boson is restricted to be of the order 100 GeV. Figure 6.1 shows the diagrams that correct this mass at one loop. From fig. 6.1a we get

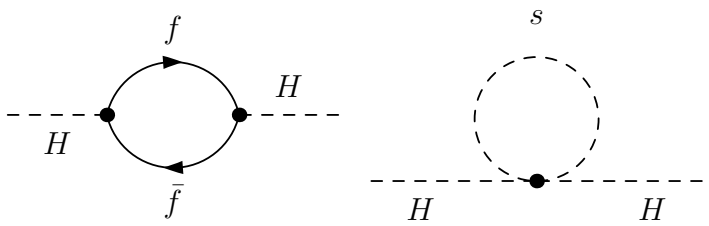


Figure 6.1: These diagrams are the one loop corrections to the Higgs mass for fermions and scalars.

the correction due to Dirac fermions with mass m_f [90]

$$\Delta m_H^2 = \frac{|\lambda_f|^2}{16\pi^2}\left[-2\Lambda_{UV}^2 + 6m_f^2 \ln\left(\Lambda_{UV}/m_f\right) + \dots\right], \tag{6.1}$$

where $\Lambda_{\rm UV}$ is an ultraviolet momentum cutoff at the scale of new physics and λ_f is the coupling derived from the term $-\lambda_f \overline{f} \Phi f$. This problem directly affects only the correction to the Higgs scalar boson (mass)^2 because the fermions and gauge bosons do not have the quadratic sensitivity to $\Lambda_{\rm UV}$. If $\Lambda_{\rm UV}$ is of the order of M_P then this correction is about 30 orders of magnitude larger than the expected mass-squared of the Higgs boson. This will indirectly affect the SM particles and their known masses as corrections to these masses have some dependence on m_H. Therefore, this cannot be the case.

The Higgs mass also receives a contribution from fig. 6.1b from the scalar particles. For a real scalar field, this contribution takes the form [90]

$$\Delta m_H^2 = \frac{\lambda_S}{16\pi^2}\left[\Lambda_{UV}^2 - 2m_S^2 \ln\left(\Lambda_{UV}/m_S\right) + \dots\right]. \tag{6.2}$$

Due to the opposite signs in (6.2) versus (6.1) there could be a particular combination of particles and masses that exactly cancel the divergences. Though plausible, it seems

extremely unlikely that all of the parameters that are currently in the SM plus any particles occurring with a higher mass are tuned exactly to cancel this.

Instead a systematic cancellation of the divergences can occur. We see that if we have two complex scalar fields for each fermion field which couple to the Higgs boson exactly as $\lambda_S = |\lambda_f|^2$ then they exactly cancel the quadratic terms. This symmetry between fermions and bosons is known as supersymmetry [90–92].

Of course this symmetry must be broken because we do not observe bosonic states of the same mass as a fermion partner. The scale at which it is broken is M_{SUSY} and the SUSY particles will have masses of the order of M_{SUSY}. In SUSY models [90] this value is expected to be at most 1 TeV in order to allow for a Higgs VEV which will give the correct values of m_Z and m_W.

At a scale of 1 TeV, the experiments at LHC should be able to discover SUSY particles. Research is still ongoing into new ways to use the general purpose experiments ATLAS [93] and CMS [94] to search for SUSY particles (for example [95, 96]).

6.1.1 Superpotential

The fermion and boson that are supersymmetric partners are combined into a multiplet called a supermultiplet. There are two kinds of supermultiplets, a chiral and a gauge supermultiplet. A chiral supermultiplet is a supermultiplet formed by the matter fields. The gauge fields and their superpartners form gauge supermultiplets. These are the objects that enter into the supersymmetric Lagrangian. This in turn can be written in terms of the *superpotential*, which will be described in this section.

It can be shown that the most general set of renormalizable SUSY interactions for a

field ψ_i and its SUSY partner can be written as

$$\mathcal{L}_{\text{int}} = -\frac{1}{2}W^{ij}\psi_i\psi_j + W^i F_i + \text{c.c.}, \tag{6.3}$$

where the W's are explained below, the F is an auxiliary term and the c.c. indicates complex conjugate. The auxiliary terms are used as a bookkeeping device and are eliminated by the equations of motion. Doing so shows that $F_i = -W_i^*$.

We can now define the superpotential as

$$W = \frac{1}{2}M^{ij}\phi_i\phi_j + \frac{1}{6}y^{ijk}\phi_i\phi_j\phi_k, \tag{6.4}$$

where M^{ij} is a mass matrix for the fermions and y^{ijk} is a Yukawa coupling of scalar, ϕ_k, and fermions ψ_i, ψ_j. This is related to the W's in eq. (6.3) by

$$W^{ij} = \frac{\delta^2 W}{\delta\phi_i\delta\phi_j} \tag{6.5}$$

$$W_i = \frac{\delta W}{\delta\phi_i}. \tag{6.6}$$

By replacing the F terms and adding the kinetic term, we find the complete Lagrangian for a chiral supermultiplet is

$$\mathcal{L}_{\text{chiral}} = \mathcal{L}_{\text{kin}} - \frac{1}{2}\left(W^{ij}\psi_i\psi_j + W^{*ij}\psi_i^\dagger\psi_j^\dagger\right) - V(\phi,\phi^*), \tag{6.7}$$

where $V(\phi,\phi^*) = W^i W_i^*$ is the scalar potential and $\mathcal{L}_{\text{kin}}$ is the standard kinetic terms for a fermion and a scalar.

The gauge field interactions also have to be written in terms of their superpartners.

For a gauge boson, A^a_μ, and its superpartner fermion, λ^a, this is

$$\mathcal{L}_{\text{gauge}} = -\frac{1}{4}F^a_{\mu\nu}F^{a\mu\nu} - i\lambda^{\dagger a}\overline{\sigma}^\mu D_\mu \lambda^a + \frac{1}{2}D^a D^a, \tag{6.8}$$

where $F^a_{\mu\nu}$ is the same term as in the SM and D^a is another auxiliary field. σ^μ and $\bar{\sigma}^\mu$ are given by

$$\sigma^\mu = (\mathbf{1}, \sigma^i), \quad \bar{\sigma}^\mu = (\mathbf{1}, -\sigma^i), \tag{6.9}$$

and are related to the standard γ-matrices by

$$\gamma^\mu = \begin{pmatrix} 0 & \sigma^\mu \\ \bar{\sigma}^\mu & 0 \end{pmatrix}. \tag{6.10}$$

It can be shown that there are two other terms that can be added and the Lagrangian will remain invariant under supersymmetric transformations. When added to the Lagrangian they give

$$\mathcal{L} = \mathcal{L}_{\text{chiral}} + \mathcal{L}_{\text{gauge}} - \sqrt{2}g\left[(\phi^* T^a \psi)\,\lambda^a + \lambda^{\dagger a}\left(\psi^\dagger T^a \phi\right)\right] + g\,(\phi^* T^a \phi)\,D^a, \tag{6.11}$$

where T^a is the generator of the group λ^a is a mediator of. It is from this equation that we can find the equations of motion to remove D^a. We find that $D^a = -g\,(\phi^* T^a \phi)$ and the $\frac{1}{2}D^a D^a$ from (6.8) combines with the last term in (6.11). Since both D^a and F_i can be expressed purely in terms of scalar fields they can be used to write the complete scalar potential as

$$V(\phi^*, \phi) = F_i^* F^i + \frac{1}{2}\sum_a D^a D^a = -W_i^* W^i + \frac{1}{2}\sum_a g_a^2\,(\phi^* T^a \phi)^2\,. \tag{6.12}$$

Fermion	Boson	$SU(3)_c$	$SU(2)_L$	$U(1)_Y$
Q_L	$\tilde{Q}_L$	**3**	**2**	$\frac{1}{6}$
u_R	$\tilde{u}_R$	**3**	**1**	$\frac{2}{3}$
d_R	$\tilde{d}_R$	**3**	**1**	$-\frac{1}{3}$
ℓ	$\tilde{\ell}$	**1**	**2**	$-\frac{1}{2}$
e_R	$\tilde{e}_R$	**1**	**1**	-1
$\tilde{H}_1$	H_1	**1**	**2**	$\frac{1}{2}$
$\tilde{H}_2$	H_2	**1**	**2**	$-\frac{1}{2}$
$\tilde{g}$	g^μ	**8**	**1**	0
$\tilde{W}$	W^μ	**1**	**3**	0
$\tilde{B}$	B^μ	**1**	**1**	0

Table 6.1: The interaction states of MSSM and the respective gauge charges. There are also three generations of (s)quarks and (s)leptons.

6.1.2 Minimal Supersymmetric Standard Model

The Minimal Supersymmetric Standard Model (MSSM) is the model that contains all of the possible $N = 1$ SUSY interactions and special cases of this model are generally analyzed phenomenologically. This model has one superpartner for every SM particle and an extra $SU(2)_L$ Higgs doublet. The extra Higgs doublet is due to the requirement that the superpotential be an analytic function. This prevents the addition of a Yukawa term like $\bar{Q}\phi^\dagger u_R$ and instead a second Higgs doublet is needed. Table 6.1 shows the particle content in terms of the interaction eigenstates. We also introduce the convention that all superpartners of fermions are names with the letter *s* preceding the name and all superpartners of bosons have *ino* appended to the name. For example the partner of an electron is a selectron and the partner of a B is a Bino. Also, by convention the symbol for a superpartner field is to put a tilde on top of the field symbol, e.g. $B^\mu \rightarrow \tilde{B}$. Bosons with a superscript μ are vector bosons, whereas those without are scalar bosons.

Just like in the SM, several of the interaction states mix to form the mass states. Table 6.2 shows which interaction states mix to form which mass states; the mass states are denoted by the word for the physical particles that are expected to be observed. The

Mixing Fermions	Mass States	Mixing Bosons	Mass States
Q_L, u_R, d_R	quarks	$\tilde{Q}_L, \tilde{u}_R, \tilde{d}_R$	squarks
ℓ, e_R	electrons, neutrinos	$\tilde{\ell}, \tilde{e}_R$	selectrons,sneutrino
$\tilde{B}, \tilde{W}_3, \tilde{H}_1, \tilde{H}_2$	neutralinos, $\tilde{\chi}_i^0$	B^μ, W_3^μ	Z^0, photon
$\tilde{W}_1, \tilde{W}_2, \tilde{H}_1, \tilde{H}_2$	charginos, $\tilde{\chi}_i^\pm$	W_1^μ, W_2^μ	$W^\pm$
		H_1, H_2	CP-even/odd Higgs, Charged Higgs
$\tilde{g}$	gluino	g^μ	gluon

Table 6.2: The interaction states that mix and yield the mass states. The mass states are given by name.

mixture of the (s)quarks given in the table is to signify that the left and right handed (s)quarks mix, not the up- and down-type (s)quarks. The Higgs sector isn't quite as straightforward. Each Higgs doublet contains a charged part and a neutral part. The real parts of the two neutral Higgs doublets mix to form the CP even Higgs states, often denoted by h and H. Here the lower case is the lighter Higgs, the one that is expected in the SM. The imaginary parts of the neutral Higgs doublets mix to form the CP odd Higgs, A. The remaining charged parts of both doublets then mix to form two charged Higgs bosons. Also, often the convention H_u and H_d is used [90]. By the definitions given here this is equivalent to $H_1 \equiv H_u$ and $H_2 \equiv H_d$. The mass states of the four neutralinos are denoted by $\chi^0_{1,2,3,4}$ in order of mass. The four mass state charginos are denoted by $\chi^\pm_{1,2}$.

6.1.2.1 Soft Breaking

As none of the superpartners of the SM particles have been observed this implies that supersymmetry must be a broken symmetry. If we refer back to the fine tuning argument we see that in order for the quadratically divergent parts to still cancel, the dimensionless couplings must cancel (i.e. $\lambda_S = |\lambda_f|^2$). This leads us to only consider "soft" breaking

of supersymmetry. This means that we can write the Lagrangian as

$$\mathcal{L} = \mathcal{L}_{\text{SUSY}} + \mathcal{L}_{\text{soft}} \tag{6.13}$$

where $\mathcal{L}_{\text{SUSY}}$ are all the terms preserving supersymmetry and $\mathcal{L}_{\text{soft}}$ are all the terms that softly break supersymmetry. To ensure renormalizability and to maintain the natural cancellation of quadratic divergences, $\mathcal{L}_{\text{soft}}$ must have only mass terms and couplings with *positive* mass dimension.

It is the existence of the soft term that allows all of the superpartners of the SM particles to be heavier than the top quark. In the absence of electroweak symmetry breaking, all of the SM fields would be massless. This isn't true for the superpartners. The scalars can have a mass term in the Lagrangian of the form $m^2 |\phi|^2$. The gauginos and Higgsinos also do not require electroweak breaking in order to acquire a mass due to the fact that they are fermions in a real representation of their gauge group.

If the largest mass term occurring in $\mathcal{L}_{\text{soft}}$ is m_{soft}. We can see that the corrections to the Higgs mass-squared, due to the SUSY particles, must vanish as $m_{\text{soft}} \to 0$. This means that these corrections cannot be quadratic in Λ_{UV} and the corrections must be of the form

$$\Delta m_H^2 = m_{\text{soft}}^2 \left[\frac{\lambda}{16\pi^2} \ln \left(\Lambda_{UV} / m_{\text{soft}} \right) + \dots \right]. \tag{6.14}$$

If we take λ of the order 1 and Λ_{UV} of order M_P, we find that the lightest SUSY particles should be about 1 TeV, as stated previously. This is what leads to the optimism that SUSY will be discovered at the LHC.

The actual soft breaking terms added to MSSM are

$$\mathcal{L}_{\text{soft}} = -\frac{1}{2} \left(M_\lambda \lambda^{\dagger a} \lambda^a + \text{c.c.} \right) - \left(m^2 \right)_j^i \phi^{*j} \phi_i - \left(\frac{1}{2} b^{ij} \phi_i \phi_j + \frac{1}{6} a^{ijk} \phi_i \phi_j \phi_k + \text{c.c.} \right). \tag{6.15}$$

There is another term that could be added without losing the renormalizability of the theory. This term is $-\frac{1}{2}c_i^{jk}\phi^{*i}\phi_j\phi_k + \text{c.c.}$ but this is often not included as it can lead to quadratic divergences in some models. It is also evident that the terms in (6.15) do break supersymmetry because they are not expressions of the supermultiplets. This is because mass terms for the fermions can be reabsorbed into a redefinition of the superpotential and the $(m^2)^i_j$ and c_i^{jk} terms.

To complete the definition of the MSSM we must also specify the superpotential. This is given as

$$W_{\text{MSSM}} = Y^u_{ij}\overline{u}H_1Q - Y^d_{ij}\overline{d}H_2Q - Y^e_{ij}\overline{e}H_2\ell + \mu H_1H_2. \tag{6.16}$$

Here the fields $Q, \ell, H_1, H_2, \overline{u}, \overline{d}, \overline{e}$ are the chiral supermultiplets, not just the SM fields. We can see the family Yukawa matrices and the Higgs fields which are used to give the masses. We also see the supersymmetric equivalent to the μ term in the SM.

This section has been a simple introduction to SUSY models, SUSY Lagrangians and MSSM. For a more detailed discussion of the development of all these topics see [90–92]. Next the concept of the effective potential and its uses is given and finally the software Effective is discussed which uses the effective potential to study $N = 1$ SUSY models.

6.2 Effective Potential

The effective potential was originally introduced by Euler and Heisenberg [97] and further expanded by Schwinger [98]. This was later applied to studies of spontaneous symmetry breaking by Goldstone, Salam and Weinberg [99]. The development of the effective potential given in this section, as well as a complete summary of the effective potential and its uses is given in [100].

The effective potential is capable of providing quite a lot of information about a theory, while working with a simpler expression. Here is presented the development of the effective potential and in subsequent sections the various things that can be derived from an effective potential are given. These are all capabilities of the software, Effective, which is discussed in the last section.

We start by giving the theoretical derivation of the effective potential. With this derivation it can be shown that the the one-loop contribution to the effective potential can be derived in a model independent way. Unfortunately, the higher order corrections cannot be developed in a model independent way. Because of the model dependence, the two-loop effective potential is not implemented in Effective, though future extensions of the code could provide this functionality.

6.2.1 Generating Functionals

As an example we start with a theory described by the scalar field ϕ with a Lagrangian density $\mathcal{L}\{\phi(x)\}$. The action is then

$$S[\phi] = \int d^4x \mathcal{L}\{\phi(x)\}. \tag{6.17}$$

The generating functional is the vacuum-to-vacuum expectation value $\langle 0_{out}|\ 0_{in}\rangle_j$ and is given by the path-integral representation,

$$Z[j] = \langle 0_{out}|\ 0_{in}\rangle_j \equiv \int \mathcal{D}\phi \exp\{i\,(S[\phi] + \phi j)\}, \tag{6.18}$$

where

$$\phi j = \int d^4x \phi(x) j(x). \tag{6.19}$$

Using (6.18) we define

$$Z[j] \equiv \exp\left\{iW\left[j\right]\right\}, \tag{6.20}$$

where $W\left[j\right]$ is known as the connected generating functional. The effective action $\Gamma\left[\bar{\phi}\right]$ is just the Legendre transformation of (6.20) and is given by

$$\Gamma\left[\bar{\phi}\right] = W[j] - \int d^4x \frac{\delta W\left[j\right]}{\delta j(x)} j(x), \tag{6.21}$$

where

$$\bar{\phi}(x) = \frac{\delta W\left[j\right]}{\delta j(x)}. \tag{6.22}$$

$\bar{\phi}(x)$ is the weighted average of the fluctuations of the field. In a translationally invariant theory, which are the ones Effective is designed to deal with, $\bar{\phi}(x)$ is a constant

$$\bar{\phi}(x) = \phi_c. \tag{6.23}$$

The effective potential can then be defined as

$$\Gamma\left[\phi_c\right] = -\int d^4x V_{\text{eff}}\left(\phi_c\right), \tag{6.24}$$

which can be written as an expansion as

$$V_{\text{eff}}\left(\phi_c\right) = -\sum_{n=0}^{\infty} \frac{1}{n!} \phi_c^n \Gamma^{(n)}\left(p_i = 0\right), \tag{6.25}$$

where $\Gamma^{(n)}$ are the one-particle irreducible (1PI) Green functions. Minimizing the effective potential over the constant fields, ϕ_c, gives the vacuum state of the theory [3].

6.2.2 One-Loop Potential

The tree-level effective potential is identical to the classical effective potential. This is simply

$$V_{\text{tree}} = -\mathcal{L}|_{\phi_i(x)\to\phi_{ic}}\,. \tag{6.26}$$

The one-loop contribution can be written in closed form for any theory containing fields of spin $0, \frac{1}{2}$, or 1.

We show here the one-loop correction for a model with one self-interacting scalar field described by the Lagrangian

$$\mathcal{L} = \frac{1}{2}\partial^\mu\phi\partial_\mu\phi - V_0(\phi), \tag{6.27}$$

where V_0 is the tree-level potential given by

$$V_0 = \frac{1}{2}m^2\phi^2 + \frac{\lambda}{4!}\phi^4. \tag{6.28}$$

As we expressed in (6.25), the one-loop correction to the tree-level effective potential is given by the sum of all 1PI diagrams with a single loop and zero external momenta. The n-th diagram has n propagators, n vertices and $2n$ external legs. The n propagators contribute a factor of $i^n(p^2-m^2+i\epsilon)^{-n}$, as we saw in chapter 1. Each pair of the external lines contributes a factor of ϕ_c^{2n} and each vertex a factor of $-i\lambda/2$. There is also a global symmetry factor of $1/2n$.

This gives

$$\begin{aligned} V_1(\phi_c) &= i\sum_{n=1}^{\infty}\int\frac{d^4p}{(2\pi)^4}\frac{1}{2n}\left[\frac{\lambda\phi_c^2/2}{p^2-m^2+i\epsilon}\right]^n \\ &= -\frac{i}{2}\int\frac{d^4p}{(2\pi)^4}\log\left[1-\frac{\lambda\phi_c^2/2}{p^2-m^2+i\epsilon}\right]. \end{aligned} \tag{6.29}$$

In (6.20) we did not include a normalization factor of $\exp\{iW[0]\}$. This is needed to correctly generate (6.29). Without it a shift is introduced to the effective potential. After a Wick rotation and using the unnormalized generating functional we find

$$V_1(\phi_c) = \frac{1}{2}\int \frac{d^4p}{(2\pi)^4} \log\left[p^2 + m^2(\phi_c)\right], \tag{6.30}$$

where the momentum is now in Euclidean space and the mass is the shifted mass given by

$$m^2(\phi_c) = \frac{d^2V_0(\phi_c)}{d\phi_c^2}. \tag{6.31}$$

This is finally

$$V_1(\phi_c) = \frac{1}{64\pi^2} m^4(\phi_c)\left(\ln\frac{m^2(\phi_c)}{\mu^2} - \frac{3}{2}\right), \tag{6.32}$$

in the $\overline{\text{DR}}$ renormalization scheme [101] where μ is the renormalization scale. This result can be trivially generalized to the case of N_s complex scalar fields each described by the Lagrangian,

$$\mathcal{L} = \partial^\mu\phi^a\partial_\mu\phi_a^\dagger - V_0(\phi^a, \phi_a^\dagger). \tag{6.33}$$

The one-loop contribution in the $\overline{\text{DR}}$ renormalization scheme is

$$V_1(\phi^a, \phi_a^\dagger) = \frac{1}{64\pi^2}\text{Tr}\left[M_s^4(\phi^a, \phi_a^\dagger)\left(\ln\frac{M_s^2(\phi^a, \phi_a^\dagger)}{\mu^2} - \frac{3}{2}\right)\right], \tag{6.34}$$

where M_s is the mass matrix of the fields given by

$$\left(M_s^2\right)_b^a = \frac{\partial^2 V}{\partial\phi_a^\dagger\partial\phi^b}. \tag{6.35}$$

Eq. (6.34) can be generalized to fermions obeying the Dirac equation and gauge

bosons [100, 102] as

$$V_1 = \sum_s \frac{1}{64\pi^2}(2s+1)(-1)^{2s}\mathrm{Tr_s}\left[M^4(\phi^a,\phi_a^\dagger)\left(\ln\frac{M^2(\phi^a,\phi_a^\dagger)}{\mu^2}-\frac{3}{2}\right)\right], \tag{6.36}$$

where the sum is over the spins, $0, \frac{1}{2}$, and 1, and Tr_s is the trace is over the mass matrix of spin, s. Defining the "Supertrace" as

$$\mathrm{STr}f(X) = \sum_i (2s_i+1)(-1)^{2s_i} f(X_i), \tag{6.37}$$

which is the spin-weighted trace, we have the one-loop contribution in the $\overline{\mathrm{DR}}$ renormalization scheme as

$$V_1 = \frac{1}{64\pi^2}\mathrm{STr}\left[M^4\left(\ln\frac{M^2}{\mu^2}-\frac{3}{2}\right)\right]. \tag{6.38}$$

6.3 Mass Matrices

The fields defined as representations of the groups in a field theory are interaction eigenstates of the theory. These are states in which the interactions are directly governed by the respective couplings. The particles that are observed are the mass eigenstates of the theory. These states are a mixture of interaction eigenstates, and thus the interactions of these states are a mixture of the couplings in the theory. The interaction states that mix to form the mass states, as well as the mixing angles, can be derived from the mass matrices. In should be noted that the mass matrices are not derivable from the effective potential. Rather, the value of the VEVs chosen from the minimization of the effective potential determine the mass spectrum of the particles. Derivation of the masses requires the full potential.

For complex scalars the mass matrix is defined as

$$\left(M_s^2\right)_b^a = \left.\frac{\partial^2 V}{\partial\phi_a^\dagger \partial\phi^b}\right|_{\phi_i(x)\to\phi_{ic}}, \tag{6.39}$$

where ϕ_{ic} is the expectation value of the the scalar field $\phi_i(x)$. Similarly for vectors the mass matrix is

$$\left(M_V^2\right)_b^a = \left.\frac{\partial^2 V}{\partial\mathcal{A}_a \partial\mathcal{A}^b}\right|_{\phi_i(x)\to\phi_{ic}}. \tag{6.40}$$

The mass matrix for fermions is not defined as a mass squared matrix. Instead for complex fermions it is given by

$$\left(M_f\right)_b^a = \left.\frac{\partial^2 V}{\partial\psi_a^\dagger \partial\psi^b}\right|_{\phi_i(x)\to\phi_{ic}}. \tag{6.41}$$

Diagonalizing these mass matrices gives

$$M_{s,V}^2 = U^T D_{s,V}^2 V, \qquad M_f = U^T D_f V, \tag{6.42}$$

where U and V are unitary matrices which define the mixing angles of the interaction states. D is the diagonal matrix whose elements are the masses of the mass eigenstates.

Eqs. (6.39-6.41) don't indicate whether the potential is to be taken at tree-level, one-loop or some other order. That is because the masses also have higher order corrections. If the tree-level potential is used, it yields the tree-level mass matrices. Likewise, if the one-loop potential is used it yields the one-loop mass matrix. For the exact mass term the exact potential would be needed.

As we saw earlier, the one-loop effective potential can be written in terms of the mass matrices only. The matrices in (6.38) are the tree level matrices. From (6.39-6.41) we can see that the one-loop masses are derived from the full one-loop potential. This would require the calculation of the full one-loop potential, which is much more complicated.

Instead the one-loop corrections to the mass matrices can be computed in another way. If we refer back to chapter 1, the masses of a field are related to the two-point Green's function of that field. In fact, it can be shown that the one-loop correction to the mass is just the one-loop correction to the two-point Green's function [3].

Before we present the one-loop corrections to the mass matrices, the idea of a ghost field must be explained. We saw that in chapter 1 that the propagator for a gauge boson is given as $\frac{-ig_{\mu\nu}}{p^2+i\epsilon}$. This is not quite true for non-Abelian groups. Instead using the Faddeev-Popov method to quantize the field and the gauge condition

$$G(A) = \partial^\mu A^a_\mu(x) - \omega^a(x) = 0, \tag{6.43}$$

where ω^a is a Gaussian weight, we find the propagator is

$$\left\langle A^a_\mu(x) A^a_\nu(x) \right\rangle = \frac{-i}{p^2 + i\epsilon}\left(g_{\mu\nu} - (1-\xi)\frac{p_\mu p_\nu}{p^2}\right). \tag{6.44}$$

The Faddeev-Popov quantization inserts a determinant of the form $\det\left(\frac{\delta G(A^\alpha)}{\delta\alpha}\right)$ where α is now used to represent the infinitesimal form of the gauge transformation. This determinant is zero for an Abelian group but non-zero for a non-Abelian group. Instead Faddeev and Popov showed it can be expressed as

$$\det\left(\frac{\delta G(A^\alpha)}{\delta\alpha}\right) = \det\left(\frac{1}{g}\partial^\mu D_\mu\right) = \int \mathcal{D}c\,\mathcal{D}\bar{c}\,\exp\left[i\int d^4x\,\bar{c}\,(-\partial^\mu D_\mu)\,c\right], \tag{6.45}$$

where c and $\bar{c}$ are now anti-commuting fields which are scalars under Lorentz transformations. These new fields are known as ghost fields and introduce new Feynman rules that must be considered to ensure that the value of an S-matrix calculation is gauge-invariant.

For a spontaneously broken symmetry, we impose the gauge condition

$$G(A) = \frac{1}{\sqrt{\xi}}(\partial_\mu A^\mu - \xi g \upsilon \phi) = 0, \tag{6.46}$$

where ϕ is the Goldstone boson associated with the breaking and υ is the VEV of the field which leads to the breaking. Using this form of the gauge fixing condition gives the massive gauge boson the propagator

$$\langle A_\mu A_\nu \rangle = \frac{-i}{p^2 - m^2}\left[g_{\mu\nu} - \frac{p_\mu p_\nu}{p^2 - \xi m^2}(1-\xi)\right], \tag{6.47}$$

which is the same value for the gauge boson propagator shown before for the massless case. This also gives the Goldstone boson, ϕ, a mass of ξm_A^2 where m_A is the mass of the gauge boson. The ghost also acquires the same mass, ξm_A^2, and this must be taken into account when computing Feynman rules of a process.

The one-loop corrections for a field can be given in a gauge independent form by explicitly keeping ξ. That has been done for the one-loop corrections given here. In Effective, the ghost contributions are hard to automate. Instead we work in the Landau gauge, $\xi \to 0$ where the masses of the Goldstone bosons remain zero and the ghost terms do not contribute to the calculations.

The one-loop corrections can be written in terms of the Passarino-Veltman [103–105] functions. These are defined by

$$\begin{aligned}
A_0(m^2;\mu^2) &= 16\pi^2\mu^{4-d}\int \frac{id^dq}{(2\pi)^d}\frac{1}{q^2 - m^2 + i\epsilon}, \\
B_0(p^2, m_1^2, m_2^2;\mu^2) &= 16\pi^2\mu^{4-d}\int \frac{id^dq}{(2\pi)^d}\frac{1}{[q^2 - m_1^2 + i\epsilon]\,[(q-p)^2 - m_2^2 + i\epsilon]}, \\
p_\mu B_1(p^2, m_1^2, m_2^2;\mu^2) &= 16\pi^2\mu^{4-d}\int \frac{id^dq}{(2\pi)^d}\frac{q_\mu}{[q^2 - m_1^2 + i\epsilon]\,[(q-p)^2 - m_2^2 + i\epsilon]}, \\
p_\mu p_\nu B_{11}(p^2, m_1^2, m_2^2;\mu^2) &+ g_{\mu\nu}B_{00}(p^2, m_1^2, m_2^2;\mu^2)
\end{aligned} \tag{6.48}$$

$$= 16\pi^2\mu^{4-d}\int\frac{id^dq}{(2\pi)^d}\frac{q_\mu q_\nu}{[q^2-m_1^2+i\epsilon]\,[(q-p)^2-m_2^2+i\epsilon]},$$

where μ[a] is the renormalization scale and the integrals are regularized in $d = 4-2\epsilon$ dimensions. The first two of these can be integrated and expressed (expanding to $\mathcal{O}(\epsilon)$) as

$$A_0(m^2;\mu^2) = m^2\left(\frac{1}{\hat{\epsilon}}-1+\ln\frac{m^2}{\mu^2}\right)+\mathcal{O}(\epsilon), \tag{6.49}$$

$$B_0(p^2,m_1^2,m_2^2;\mu^2) = \frac{1}{\hat{\epsilon}}-\ln\frac{p^2}{\mu^2}-f_B(x_+)-f_B(x_-)i+\mathcal{O}(\epsilon), \tag{6.50}$$

where $1/\hat{\epsilon} = 1/\epsilon-\gamma_E+\ln 4\pi$,

$$x_\pm = \frac{s\pm\sqrt{s^2-4p^2(m_1^2-i\epsilon)}}{2p^2},\quad f_B(x) = \ln(1-x)-x\ln\left(1-\frac{1}{x}\right)-1, \tag{6.51}$$

and $s = p^2-m_2^2+m_1^2$. The other Passarino-Veltman functions can be decomposed into the two scalar functions, A_0 and B_0 [104]. There are more functions for higher loop corrections that are not given here. These can be decomposed into their respective higher-loop scalar functions (e.g. C_0, D_0, ...).

The corrections are summed over all internal fields and the couplings are given from the Lagrangian. There are two ways that the one-loop corrections can be applied. First one could find the diagonal form of the tree-level matrix. This defines the mass eigenstates and therefore the couplings of these states. The corrections can then be directly applied to the tree-level masses given the couplings defined in this manner. This is not exactly the one-loop correction, however. This will give a close approximation to the one-loop masses, but will only give the tree-level mixings. If instead the one-loop corrections are applied to the undiagonalized matrix, with corrections to the off-diagonal elements, then the diagonalization of this corrected matrix will give the correct one-loop

[a]Not to be confused with μ for the Lorentz index

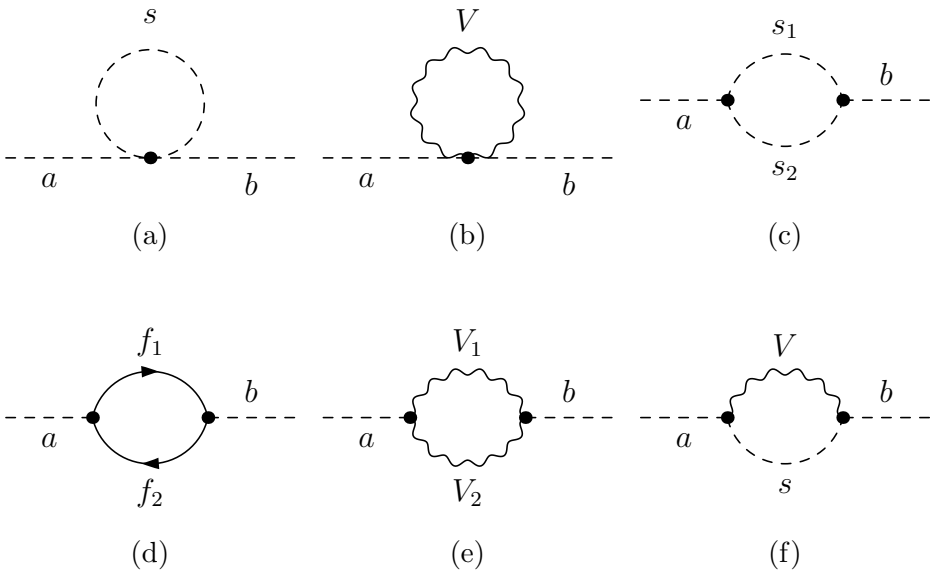


Figure 6.2: The different diagrams that contribute to the one-loop correction of the scalar mass matrix element $(M_s^2)_{ab}$.

masses and mixings. Both techniques have been implemented in Effective, but the latter is much more time intensive and if only the masses are desired, applying the corrections to the mass eigenstates is a good approximation.

6.3.1 Scalar One-Loop Corrections

We first give the one-loop corrections to the scalar masses. Figure 6.2 shows the corrections to the scalar mass matrix term $(M_s^2)_b^a$. These diagrams are summed over all possible intermediate states given in the theory. The results given here have been calculated with the help of the computer software FeynCalc [106] and FormCalc [107].

The form for the contribution of the scalar four-point vertex, given by the first diagram in figure 6.2 is

$$\left(\Sigma^{(assb)}\right)_b^a (p^2, m^2; \mu^2) = \frac{iC^{(assb)}}{32\pi^2} A_0(m^2; \mu^2), \tag{6.52}$$

where $C^{(assb)}$ is the four-point coupling of the fields a and b with an arbitrary scalar field s. The second diagram in figure 6.2 is the contribution of the four-point vertex given with a vector loop. This correction is

$$\left(\Sigma^{(aVVb)}\right)^a_b(p^2, m^2; \mu^2) = \frac{iC^{(aVVb)}}{32\pi^2}\left(2m^2 - 3A_0(m^2) + \xi A_0(\xi m^2)\right), \tag{6.53}$$

The diagram given by fig. 6.2c is of the scalar three-point vertex. This is given by

$$\left(\Sigma^{(as_1s_2b)}\right)^a_b(p^2, m_1^2, m_2^2; \mu^2) = \frac{-C^{(as_1s_2)}C^{(s_1s_2b)}}{32\pi^2}B_0(p^2, m_1^2, m_2^2) \tag{6.54}$$

where $C^{(as_1s_2)}$ is the coupling of field a to the two fields in the loop and $C^{(s_1s_2b)}$ is the coupling of field b.

There is no fermion four-point diagram as this type of term would have a mass dimension greater than four in the Lagrangian. Instead there is only a three-point correction. This term (fig. 6.3d) is given by the expression

$$\begin{aligned}\left(\Sigma^{(af_1f_2b)}\right)^a_b(p^2, m_1^2, m_2^2; \mu^2) &= \frac{C^{(af_1f_2)}C^{(f_1f_2b)}}{16\pi^2}\left\{A_0(m_1^2; \mu^2) + A_0(m_2^2; \mu^2) + \right. \\ &\left. ((m_1 + m_2)^2 - p^2)B_0(p^2, m_1^2, m_2^2; \mu^2)\right\}. \end{aligned} \tag{6.55}$$

The vector three-point term, given by the fifth diagram in figure 6.2, is

$$\begin{aligned}\left(\Sigma^{(aV_1V_2b)}\right)^a_b(p^2, m_1^2, m_2^2; \mu^2) &= \frac{C^{(aV_1V_2)}C^{(V_1V_2b)}}{32\pi^2}\left\{2 + (\frac{p^2}{m_2^2} - 3)B_0^0 - \frac{p^2}{m_2^2}B_0^2 + \right. \\ &\xi\frac{A_0^2}{m_2^2} - \xi\frac{A_0^4}{m_2^2} + \left(\xi^2\frac{m_1^2}{m_2^2} - \xi\right)B_0^1 - \xi^2\frac{m_1^2}{m_2^2}B_0^{12} + \\ &\frac{p^2}{m_1^2m_2^2}\left(B_{00}^2 - B_{00}^0 + B_{00}^1 - B_{00}^{12}\right) + \\ &\frac{p^4}{m_1^2m_2^2}\left(B_{11}^2 - B_{11}^0 + B_{11}^1 - B_{11}^{12}\right) + \\ &\left. 2\xi\frac{p^2}{m_2^2}(B_1^1 - B_1^{12})\right\}, \end{aligned} \tag{6.56}$$

where

$$\begin{aligned} A_0^1 &\equiv A_0(m_1^2;\mu^2), \quad A_0^2 \equiv A_0(m_2^2;\mu^2), \quad A_0^3 \equiv A_0(\xi m_1^2;\mu^2), \quad A_0^4 \equiv A_0(\xi m_2^2;\mu^2), \\ B_x^0 &\equiv B_x(p^2,m_1^2,m_2^2;\mu^2), \quad B_x^1 \equiv B_x(p^2,m_2^2,\xi m_1^2;\mu^2), B_x^2 \equiv B_x(p^2,m_1^2,\xi m_2^2;\mu^2), \\ B_x^{12} &\equiv B_x(p^2,\xi m_1^2,\xi m_2^2;\mu^2). \end{aligned} \tag{6.57}$$

Remarkably, in the 't Hooft-Feynman gauge ($\xi \to 1$) this reduces to

$$\left(\Sigma^{(aV_1V_2b)}\right)_b^a(p^2,m_1^2,m_2^2;\mu^2) = \frac{C^{(aV_1V_2)}C^{(V_1V_2b)}}{16\pi^2}\left(1-2B_0(p^2,m_1^2,m_2^2;\mu^2)\right). \tag{6.58}$$

The last one-loop corrections to the scalar mass matrix, $(M_s^2)_b^a$, is given by the last diagram in figure 6.2. This is the correction given by the vector-scalar three-point vertex, given by

$$\begin{aligned} \left(\Sigma^{(aVsb)}\right)_b^a(p^2,m_V^2,m_s^2;\mu^2) &= \frac{-C^{(aVs)}C^{(Vsb)}}{16m_V^2\pi^2}\left\{(p^2-m_s^2)A_0^1-(p^2-\xi m_V^2-m_s^2)A_0^3+\right. \\ &\quad \left(m_V^2(p^2+m_s^2)-(p^2-m_s^2)^2\right)B_0^0- \\ &\quad \left.(p^2-m_s^2)^2B_0^1-2m_V^2p^2B_1^0\right\}, \end{aligned} \tag{6.59}$$

where the vector corresponds to loop field 1 in the previous abbreviations and the scalar field is loop field 2.

6.3.2 Vector One-Loop Corrections

The one-loop corrections to the vector fields can also be generated using the same two programs FeynCalc and FormCalc. Figure 6.3 shows the diagrams which give corrections to the vector mass.

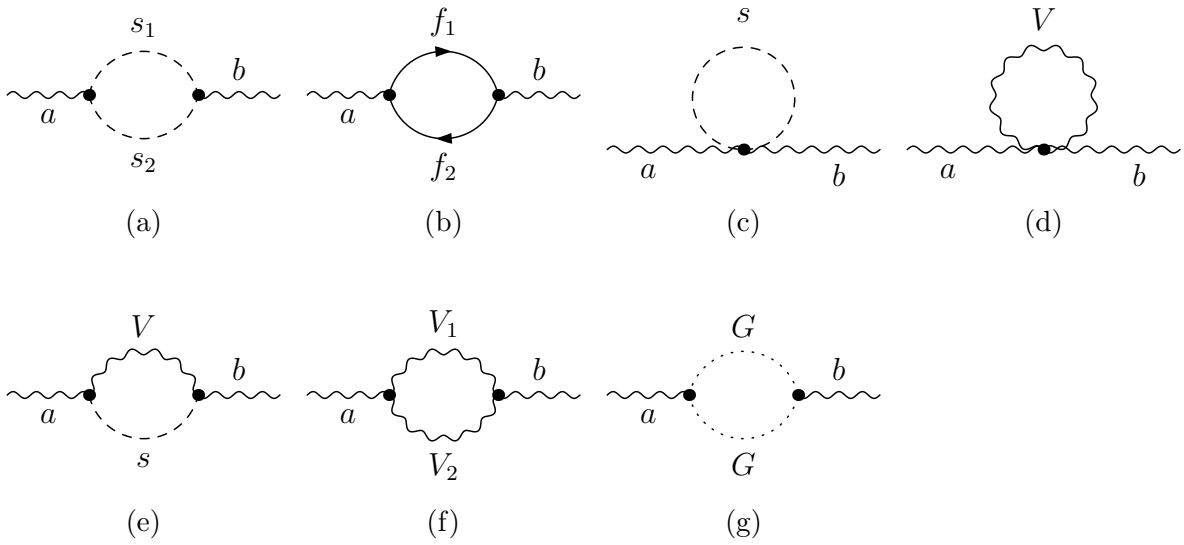


Figure 6.3: The different diagrams that contribute to the one-loop correction of the vector mass matrix element $(M_V^2)_{ab}$.

The first diagram in figure 6.3 leads to a correction of the form

$$\left(\Pi_{\mu\nu}^{(as_1s_2b)}\right)_b^a(p^2, m_1^2, m_2^2; \mu^2) = \frac{C^{(as_1s_2)}C^{(s_1s_2b)}}{8\pi^2} B_{00}(p^2, m_1^2, m_2^2; \mu^2)\left(g_{\mu\nu} - \frac{p_\mu p_\nu}{p^2}\right). \quad (6.60)$$

The contribution from the fermion loop is gauge independent and is given by

$$\begin{aligned}\left(\Pi_{\mu\nu}^{(af_1f_2b)}\right)_b^a(p^2, m_1^2, m_2^2; \mu^2) &= -\frac{C^{(af_1f_2)}C^{(f_1f_2b)}}{16\pi^2}\left\{A_0(m_1^2;\mu^2) + A_0(m_2^2;\mu^2) + \right.\\ &\quad \left((m_1-m_2)^2 - p^2\right)B_0(p^2, m_1^2, m_2^2; \mu^2) - \\ &\quad \left. 4B_{00}(p^2, m_1^2, m_2^2; \mu^2)\right\}\left(g_{\mu\nu} - \frac{p_\mu p_\nu}{p^2}\right), \end{aligned} \quad (6.61)$$

and fig. 6.3c shows the scalar four-point correction. This is

$$\left(\Pi_{\mu\nu}^{(asb)}\right)_b^a(p^2, m_1^2, m_2^2; \mu^2) = \frac{iC^{(aSSb)}}{32\pi^2} A_0(m_S^2;\mu^2)\left(g_{\mu\nu} - \frac{p_\mu p_\nu}{p^2}\right). \quad (6.62)$$

The next two diagrams are of the four-point vector loop and the three-point vector-scalar loop. These are given by

$$\left(\Pi_{\mu\nu}^{(aVVb)}\right)_b^a(p^2,m^2;\mu^2)=\frac{iC^{(aVVb)}}{32\pi^2}\left(2m^2-5A_0(m^2;\mu^2)-\xi A_0(\xi m^2;\mu^2)\right)\left(g_{\mu\nu}-\frac{p_\mu p_\nu}{p^2}\right), \tag{6.63}$$

and

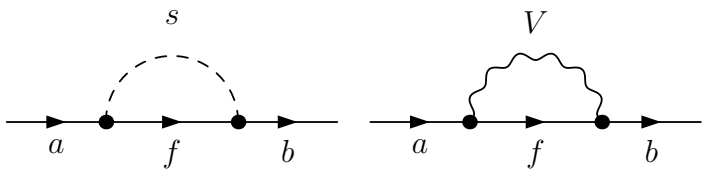
$$\left(\Pi_{\mu\nu}^{(aVsb)}\right)_b^a(p^2,m_V^2,m_s^2;\mu^2)=\frac{iC^{(aVs)}C^{(Vsb)}}{32\pi^2m_V^2}\left(m_V^2B_0^2-B_{00}^2+B_{00}^{12}\right)\left(g_{\mu\nu}-\frac{p_\mu p_\nu}{p^2}\right), \tag{6.64}$$

where we have used the same abbreviations (6.57) and the vector corresponds to field 1 in the notation and the scalar is field 2. The remaining two diagrams are for the vector three-point correction and the ghost terms. The vector three-point correction is quite complicated and is given by

$$\begin{aligned}\left(\Pi_{\mu\nu}^{(aV_1V_2b)}\right)_b^a(p^2,m_1^2,m_2^2;\mu^2) &= \frac{C^{(aV_1V_2)}C^{(V_1V_2b)}}{32\pi^2}\Big\{\frac{2}{3}p^2-2(m_1^2+m_2^2)+\\ &(\rho_2-\kappa_1+\xi)A_0^2-(\rho_2-\kappa_1-\xi)A_0^4+\\ &(m_1^2(1-\kappa_1)+p^2(2\kappa_1+5-\rho_2)B_0^0+\\ &(m_1^2(\kappa_1+\xi^2)+p^2(\rho_2-2\kappa_1))B_0^2+\\ &(\kappa_1+11-2\rho_2-5\rho_1+\rho_1\rho_2)B_{00}^0+\\ &(2\rho_2-\kappa_1-\rho_1\rho_2)B_{00}^2+(5\rho_1-\rho_1\rho_2-\xi)B_{00}^1+\rho_1\rho_2B_{00}^{12}+\\ &-2p^2B_1^0+4\xi p^2B_1^1+\\ &4\rho_1p^2\left(B_{11}^1-B_{11}^0\right)\Big\}\left(g_{\mu\nu}-\frac{p_\mu p_\nu}{p^2}\right),\end{aligned} \tag{6.65}$$

where the abbreviations (6.57) are used and $\kappa_1=\frac{m_1^2}{m_2^2}$, $\rho_1=\frac{p^2}{m_1^2}$ and $\rho_2=\frac{p^2}{m_2^2}$. In the 't Hooft-Feynman gauge this reduces to

$$\begin{aligned}\left(\Pi_{\mu\nu}^{(aV_1V_2b)}\right)_b^a(p^2,m_1^2,m_2^2;\mu^2) &= \frac{C^{(aV_1V_2)}C^{(V_1V_2b)}}{32\pi^2}\Big\{\frac{2}{3}p^2-2(m_1^2+m_2^2)+\end{aligned}$$



(a) (b)

Figure 6.4: The different diagrams that contribute to the one-loop correction of the fermion mass matrix element $(M_f)^a_b$.

$$2\left(m_1^2+3p^2\right)B_0(p^2,m_1^2,m_2^2;\mu^2)-2\kappa_1 A_0(m_2^2;\mu^2)+$$
$$(10+\rho_1)B_{00}(p^2,m_1^2,m_2^2;\mu^2)+$$
$$2p^2B_1\left(p^2,m_1^2,m_2^2;\mu^2\right)\Big\}\left(g_{\mu\nu}-\frac{p_\mu p_\nu}{p^2}\right). \qquad (6.66)$$

It must also be noted that the gauge bosons from an unbroken group are massless. These bosons only couple to themselves and the correction is proportional to p^2. The correction is to be evaluated at $p^2=m^2$, which for bosons from an unbroken group is zero.

The remaining one-loop correction to the vector mass matrix is from the ghost terms. This is the last diagram in figure 6.3 and is given by

$$\left(\Pi^{(aGGb)}_{\mu\nu}\right)^a_b(p^2,m_G^2;\mu^2)=\frac{C^{(aGG)}C^{(GGb)}}{32\pi^2}\xi B_{00}(p^2,\xi m_G^2,\xi m_G^2;\mu^2)\left(g_{\mu\nu}-\frac{p_\mu p_\nu}{p^2}\right), \quad (6.67)$$

from which it is easy to see that the ghost contribution is zero in the Landau gauge.

6.3.3 Fermion One-Loop Corrections

There are only two mass corrections to the fermion masses. The diagrams of these are given in fig. 6.4. The first one is the correction due to a scalar-fermion three-point

coupling. This is

$$\left(\Sigma^{(asfb)}\right)_b^a(p^2, m_s^2, m_f^2; \mu^2) = \frac{-C^{(asf)}C^{(sfb)}}{16\pi^2}\not{p}B_1(p^2, m_s^2, m_f^2; \mu^2). \tag{6.68}$$

The mass correction is given by $\delta_m = \Sigma(\not{p})|_{\not{p}=m}$. This can be approximated by $\delta_m \approx \Sigma(\not{p})|_{\not{p}=m_0}$.

Lastly, the correction given by fig. 6.4b is from a vector boson three-point coupling. This is

$$\left(\Sigma^{(aVfb)}\right)_b^a(p^2, m_V^2, m_f^2; \mu^2) = \frac{C^{(aVf)}C^{(Vfb)}}{8\pi^2}\not{p}\left[B_1^0 - (1-\xi)(B_{00}^1 + p^2 B_{11}^1)\right], \tag{6.69}$$

where the abbreviations given earlier are used and the index 1 refers to the vector and the index 2 is the fermion.

6.4 Renormalization Group Equations

The basic concept of renormalization stems from the observation that the divergences in the one-loop graphs amount to shifts in the parameters of the action. For example, as we saw in the last section they change the mass of a field. Renormalization is the procedure of cancelling the divergences from these shifts by introducing counterterms into the Lagrangian. These counter terms are defined such that they exactly cancel the divergent quantity in the one-loop correction but leave the finite parts untouched [5].

We can first consider the ϕ^4 theory. The Lagrangian is given by

$$\mathcal{L} = (\partial\phi_0)^2/2 - m_0^2\phi_0^2/2 - g_0\phi_0^4/4! \tag{6.70}$$

We can then rescale the field by $\phi_0 = Z^{1/2}\phi$, so that in terms of the new 'renormalized

field' ϕ we have

$$\mathcal{L} = Z(\partial\phi)^2/2 - Zm_0^2\phi^2/2 - g_0 Z^2\phi^4/4! \tag{6.71}$$

We can then write this in terms of the counterterms

$$\begin{aligned}\mathcal{L} &= (\partial\phi)^2/2 - m^2\phi^2/2 - g\phi^4/4! \\ &\quad +\delta_Z(\partial\phi)^2/2 - \delta m^2\phi^2/2 - \delta g\phi^4/4!\end{aligned} \tag{6.72}$$

We can see that $\delta_Z = Z - 1$, $\delta m^2 = Zm_0^2 - m^2$ and $\delta g = Z^2 g_0 - g$. These counterterms are adjusted to cancel the divergences of each term.

To renormalize a theory renormalization conditions must be imposed. The general conditions that are imposed are to set the contributions from the 1PI diagrams to the two-point Green's functions and the derivatives of these diagrams to zero at a spacelike momentum, $p^2 = -\mu^2$. μ is then known as the *renormalization scale*. The conditions are imposed at a spacelike momentum in order to avoid threshold singularities. Similarly, the 1PI contributions to the three- and four-point Green's functions are defined such that the coupling at that scale is exactly the physical coupling (e.g. g from (6.72)).

The renormalization scale is completely arbitrary. The theory can just as easily be defined at another renormalization scale. This implies that the theory and the redefined couplings should depend on the scale in such a way that physical calculations are independent of the scale. For example if we take an n-point Green's function we require

$$\delta G^{(n)} = \frac{\partial G^{(n)}}{\partial\mu}\delta\mu + \frac{\partial G^{(n)}}{\partial\lambda}\delta\lambda = n\delta\eta G^{(n)}. \tag{6.73}$$

This is for either a massless field or if the mass of the field is considered as on of the vertices that needs to be renormalized. $\delta\lambda$ is the counterterm corresponding to the coupling λ and $\delta\eta$ is the rescaling of the external field in the n-point Green's function.

Defining

$$\beta_\lambda \equiv \mu\frac{\partial\lambda}{\partial\mu}, \qquad \gamma_\eta \equiv -\mu\frac{\partial\eta}{\partial\mu}, \tag{6.74}$$

we can now give the Callan-Symanzik equation [108, 109]. This is

$$\left[\mu\frac{\partial}{\partial\mu} + \beta_\lambda\frac{\partial}{\partial\lambda} + n\gamma_\eta\right] G^{(n)} = 0. \tag{6.75}$$

The β and γ functions are the same between different Green's functions. Knowing this we can find the γ function from the 2-point Green's function of the field. The 2-point Green's function can be expressed in terms of the loop diagrams and the counterterms. Doing so γ can be shown to be

$$\gamma = \frac{1}{2}\mu\frac{\partial}{\partial\mu}\delta_Z. \tag{6.76}$$

The counterterm can be written in the form (at lowest order)

$$\delta_Z = A\log\frac{\Lambda^2}{\mu^2} + \text{finite}, \tag{6.77}$$

in order to cancel the divergent logarithm in $G^{(2)}$. This means that we can find γ simply by determining A, the coefficient of the divergent piece of the 1PI diagrams. Since γ is related to the field renormalization it is known as the *anomalous dimension* of the field.

Knowing the anomalous dimensions, the β functions can be derived by using the appropriate Green's function ($G^{(n)}$ for an n-point coupling). The n-point Green's function is given by the sum of the tree-level diagram, the 1PI loop diagrams of the vertex, the vertex counterterm and the external leg corrections. This can be expressed for the n-point coupling g, at one-loop order, as

$$G^{(n)} = \left(\prod_i \frac{i}{p_i^2}\right)\left[-ig - iB\log\frac{\Lambda^2}{-p^2} - i\delta_g + (-ig)\sum_i\left(A_i\log\frac{\Lambda^2}{-p_i^2} - \delta_{Z_i}\right)\right], \tag{6.78}$$

plus the finite terms. Applying the Callan-Symanzik equation we find

$$\beta_g = \mu\frac{\partial}{\partial\mu}\left(-\delta_g + \frac{1}{2}g\sum_i \delta_{Z_i}\right). \tag{6.79}$$

We can see that the second term on the right is just the sum of the anomalous dimensions of all the external particles in the Green's function. At one-loop, δ_g is given by an expression of the form

$$\delta_g = -B\log\frac{\Lambda^2}{\mu^2} + \text{finite}. \tag{6.80}$$

This means that we just have to compute the divergent pieces of the 1PI corrections to the coupling and β_g is given by

$$\beta_g = -2B + g\sum_i \gamma_i. \tag{6.81}$$

Most theories have many couplings and fields associated with them. This requires the procedure that produced (6.75) to be applied to a set of couplings, λ_I, and a set of fields ϕ_i.

$$\left[\mu\frac{\partial}{\partial\mu} + \beta_I\frac{\partial}{\partial\lambda_I} + \gamma_i\phi_i\frac{\partial}{\partial\phi_i}\right]\mathcal{O}(\lambda_I, m_i; \mu) = 0, \tag{6.82}$$

We saw earlier that using the two-, three- and four-point Green's functions, eq. (6.82) can be applied to determine the form of all β_I and all γ_i, known as the *renormalization group equations* [110–112]. This technique requires the calculation of several 1PI diagrams and is quite computationally intensive. Instead there are some couplings which can be computed either completely for an arbitrary theory, or by applying the Callan-Symanzik equation to a different set of observables.

The running of the gauge group couplings can be computed from many different vertices. They all require the boson self-energy which is given by the diagrams in fig. 6.3. The divergences of these diagrams define the anomalous dimension of the gauge

boson field. These sum to give

$$\gamma_b = \frac{g^2}{(4\pi)^2}\left[\left(\frac{13}{6}-\frac{\xi}{2}\right)C_2(G) - \frac{4}{6}\sum_i C(r_i) - \frac{1}{6}\sum_j C(r_j)\right], \tag{6.83}$$

where i is the sum over fermions, j is the sum over scalars, $C_2(G) = f^{acd}f^{acd}$ and $C(r) = \mathrm{Tr}[t_r^a t_r^a]$. One can then use the fermion-fermion-vector coupling, as in [3], or one could just as easily use the gauge boson three- or four-point couplings (in non-Abelian theories). The final result is gauge independent and given in general by the expression

$$\beta_g = -\frac{g^3}{(4\pi)^2}\left[\frac{11}{3}C_2(G) - \frac{4}{3}\sum_i C(r_i) - \frac{1}{3}\sum_j C(r_j)\right], \tag{6.84}$$

where again the index i indicates the fermions and j the scalars. For the case of Weyl spinors, the sum of the fermions has an additional factor of a half (e.g. $\frac{2}{3}$ rather than $\frac{4}{3}$), and similarly for real scalars.

The SM has Yukawa couplings, gauge couplings and couplings of the Higgs boson (λ and μ). These are all dependent on the scale [b], Q, at which they are evaluated. A common argument for beyond standard model (BSM) physics is that given the particles in the SM, the couplings, α, α_{weak} and α_{s}, will never meet at a single ultimate scale, known as the unification scale. Though there is no proof that indicates these should unify, it is believed to be a desired and expected result of a theory. Figure 6.5 shows the running of the different α's for the particles in the EW theory with one generation (black, solid), the SM with three generations (red, dashed) and the MSSM with three generations (blue, dot-dashed). We can see from this figure that the couplings from the MSSM all meet at a single scale. The derivation of the β functions and the running of the couplings in this figure were all done by entering the models into Effective. In unification theories, the $U(1)$ coupling is multiplied by $\sqrt{5/3}$ and this is also done here.

[b] Here Q is used instead of μ to avoid confusion with the Higgs coupling, μ.

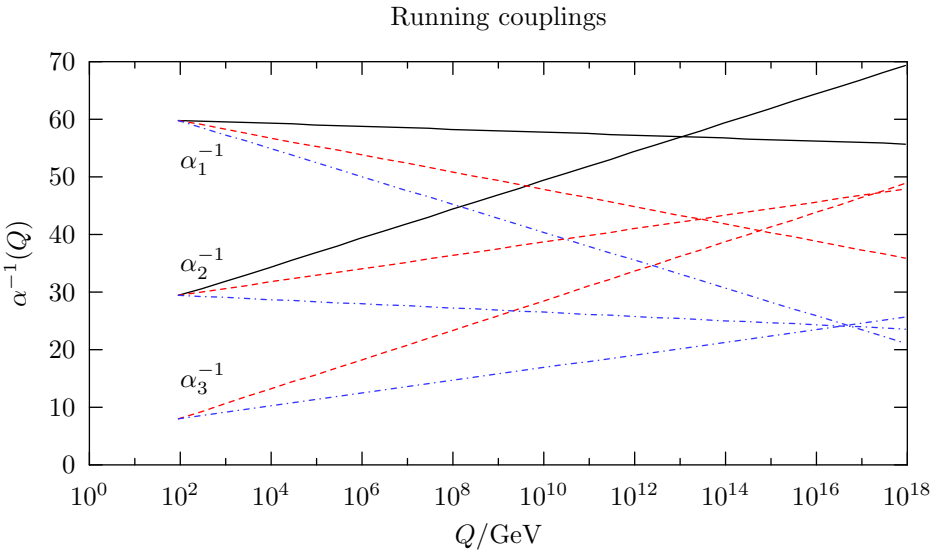


Figure 6.5: The scale dependence of the different gauge couplings. The black (solid) lines are for the EW model with one fermion generation. The red (dashed) lines are for the SM with all three fermion generations. The blue (dot-dashed) lines show the the couplings for the set of particles in the MSSM.

In Effective the observable that is used to define some of the couplings is the effective potential [100, 113]. The one-loop effective potential $V_0 + V_1$, where V_1 is given in (6.38), can be used to define the RGEs for all the couplings that appear in the potential. This is the set of couplings for the fields that may acquire a non-zero VEV. If we apply the Callan-Symanzik equation to the one-loop effective potential, where now the derivatives with respect to the fields are replaced by derivatives with respect to the VEVs, this yields

$$\left[\beta_I \frac{\partial}{\partial \lambda_I} + \gamma_i \phi_{ic} \frac{\partial}{\partial \phi_{ic}}\right] V_0 = \frac{1}{32\pi^2} \mathrm{STr}\ M^4. \tag{6.85}$$

Using this one can compare the powers of the VEVs (ϕ_{ic}) on the left hand side of the equation to those of the right and derive the relations of the β functions, given the anomalous dimensions, γ_i.

To compute the remaining β functions, we must use (6.81). This requires us to compute the 1PI diagrams for the three- and four-point vertices given in fig. 6.6. Since we have already derived the β functions of the gauge group couplings we do not need to

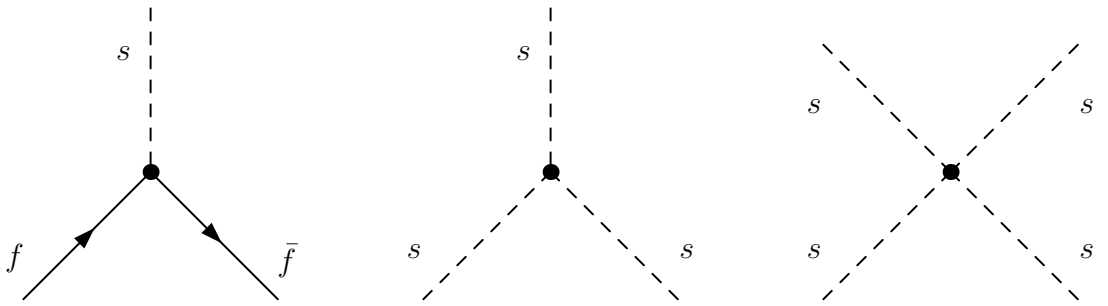


Figure 6.6: The three- and four-point diagrams that must have their 1PI diagrams computed to define the complete set of β functions for a theory.

consider the vertices with a gauge boson as an external leg.

6.5 Effective

Effective is a computer program that is able to generate the effective potential, the mass spectrum and the RGEs of a theory. It is originally based on the work in [114] and uses the GiNaC [115] computer algebra system to do algebraic manipulations. This package provides for evaluations and simplifications of complex algebraic structures within the C++ environment. As this is a C++ library, it also allows for expansion. New functions and objects can be created that can then have their own set of functions that are used during simplifications and evaluations. It is the ability to integrate the computer algebra with the standard C++ functionality and to be able to define new objects that interact with the computer algebra package that make GiNaC an ideal library for building Effective on.

My contribution to Effective covers a broad spectrum. I have restructured and written the code to work with the algebra package, GiNaC. The original one-loop mass corrections in the code were only for scalar particles. These were taken from the literature and did not all use the same gauge and as a complete correction, were wrong. As has been seen earlier, these have been rederived in an arbitrary gauge for all different spins and implemented in Effective in the Landau gauge. Also, I have done the development and implementation of the RGEs in Effective.

Effective has currently been designed to work with $N = 1$ SUSY. Effective itself is a package that can be used to define a SUSY model. Once this model is defined a user can interact with it. It is these different interactions that are discussed in this section.

6.5.1 Model Definition

A model in Effective is defined by a set of gauge groups, fields and additional interactions. These are then used to define a Lagrangian and an effective potential. The Lagrangian can be used to define the couplings of the theory and the mass matrices. As we saw in the last section, the effective potential as well as the Green's functions can define the RGEs of the theory.

A gauge group is just that. In Effective this is a class that defines the structure constants and the generators. This also defines the dimension of the indices in the fundamental and adjoint representations. It is also possible to have several gauge groups of the same group structure defined. For example if a user wants to test a theory with two $SU(2)$ groups this can be done. Each gauge group has an extra flag to indicate the 'line' that it is part of. If someone wants to use two gauges with the same group structure, they just have to define a new line for the new group. When indices are being contracted, these lines are compared and only when they are identical will a contraction

occur. Effective currently has the $SU(3)$, $SU(2)$ and $U(1)$ groups implemented, but it is easy to expand and add new groups. An example of this is given in the appendix.

Once the groups have been defined, the gauge fields can be added. These are the fields which mediate the interactions. It is possible to create gauge bosons and their superpartners, gauge fermions. When the fields are created the relevant terms can be added to the Lagrangian. This is given in section 6.1 but is repeated here for convenience

$$\mathcal{L}_{\text{gauge}} = -\frac{1}{4}F^{a\mu\nu}F^{a}_{\mu\nu} - i\lambda^{\dagger a}\bar{\sigma}^{\mu}D_{\mu}\lambda^{a}, \tag{6.86}$$

where λ^a is the gauge fermion and $F^a_{\mu\nu}$ is the standard tensor formed by a gauge boson. In $N = 1$ SUSY there are no other terms which depend only on the gauge mediators. It is possible to extend Effective to provide for $N \neq 1$ SUSY. If this were done then a new set of terms will also need to be added to the Lagrangian.

After the gauge groups and the gauge fields are defined the matter fields can be given. These are fields that can have charges under more than one group. These fields also define the other gauge invariant terms. For a given fermion, ψ, and superpartner scalar, ϕ, we have the terms

$$\mathcal{L}_{\text{matter}} = \psi^{\dagger}\bar{\sigma}^{\mu}D_{\mu}\psi + \left(D^{\mu}\phi\right)^{\dagger}D_{\mu}\phi - \sqrt{2}g_i\left[\left(\phi^{*}T^{a}\phi\right)\lambda^{a} + \text{c.c.}\right] + \frac{1}{2}g_i^2\left(\phi^{*}T^{a}\phi\right)^2, \tag{6.87}$$

where λ is again the gauge mediating fermion, D_μ is the covariant derivative for the groups the fields are charged under, and g_i is the coupling of the i-th gauge group. This expression is summed over all the gauge groups of the matter fields. It is also at this point that the VEVs of the scalar fields are defined. A set of parameters can be defined and combinations of these parameters can be given as the VEVs. For example, in MSSM one often works with the VEVs in terms of υ and $\tan\beta$, not the actual VEVs, υ_1 and υ_2. This can also be done in Effective so the set of parameters that one works with

is consistent with the set of parameters given in the literature. An example of this is given in the appendix. Unfortunately, some of the routines in Effective require the actual VEVs to be entered. These include the RGEs and the minimization of the potential. If neither of these functions is needed, however, then the set of parameters used in a paper can also be used in Effective.

Once the fields of the model are given, the superpotential can be defined. In Effective this is not given as a function of superfields, as the definition of the superpotential is, but the user must specify this in terms of the relevant fields (scalars, fermions, etc.). This is a feature of the code that could be improved to provide the appropriate superfield formalism. Once the superpotential is defined then the following terms are added to the Lagrangian

$$\mathcal{L}_{\mathrm{W}} = \left(W^{ij}\psi_i\psi_j + \mathrm{c.c.}\right) - W_i^* W^i, \tag{6.88}$$

where we saw before that

$$W^{ij} = \frac{\partial^2 W}{\partial\phi_i \partial\phi_j}, \quad W^i = \frac{\partial W}{\partial\phi_i}. \tag{6.89}$$

After the definition of the superpotential the last remaining ingredient of a model is the symmetry breaking terms. These are usually the soft breaking terms of a SUSY model but Effective can be used to work with non-SUSY models as well. In this type of model one would define any other gauge invariant terms that are desired for the model but have not been added up to this point (e.g. the Yukawa terms).

6.5.2 Mass Matrices

Once the VEVs have been defined the code can generate the tree-level mass matrices. Effective is able to do this and produce the matrices in analytic form. It is important to

note that in Effective the real and imaginary parts of the fields are treated independently. The separation of these parts has been implemented to provide a mechanism for studying CP violating models. This separation causes several complications when comparing to the literature though.

Often in the literature the states are given as the mass eigenstates. These are a combination of real and imaginary parts of the the interaction states. It is these real and imaginary combinations that must be compared to the results of Effective for a true comparison. Often this is a tedious task. Instead it is usually more convincing, and easier, to check the numerical results of Effective against the literature.

Another complication that arises when implementing formulae in Effective is that Weyl spinors are used, not Dirac ones which are often used in literature. This affects the degrees of freedom of the fields in Effective. Usually one considers a fermionic field to have a spin factor of -2 from the spin coefficient $(2s+1)(-1)^{2s}$. Instead since Effective uses Weyl spinors, this has a factor of $\frac{1}{2}$ attached to it, giving -1. The difference between Dirac spinors and Weyl spinors must also be taken into account when implementing the one-loop corrections given in section 6.3.

As mentioned earlier, Effective uses the Landau gauge ($\xi = 0$) to avoid the need for ghost contributions. The one-loop corrections to the mass matrices have been implemented in Effective in this gauge. These corrections are given in numerical form only. There are two ways in which the corrections can be applied. The first is that the tree-level matrices are diagonalized and the mixing matrices are set. The corrections are then applied directly to the mass eigenstates. This is only an approximation to the one-loop corrections and provides only the tree-level mixing matrices. The other way in which the one-loop corrections can be applied is to add the one-loop corrections of the interaction eigenstates to the tree-level mass matrix. This new matrix is then diagonalized and this process defines the one-loop mixing matrices and the one-loop masses. Effective pro-

duces the tree-level matrices and the corresponding diagonal form in order to perform all its other calculations. The full one-loop correction requires performing an additional diagonalization for each set of parameters one would want to compute the masses at. Since the diagonalization of the matrices is one of the most CPU intensive calculations performed by Effective, doing the full one-loop corrections is very time consuming.

6.5.3 Effective Potential

At this point the model has been completely defined and the tree-level masses have been computed. It is now possible to find the effective potential and minimize it over a set of parameters. The analytic tree-level effective potential is generated by Effective and can be returned. Effective also provides a function `ex extremizePotential(vector<numeric*>)` that goes through the parameters given in the input vector and minimizes the potential over these parameters using the Powell [11] minimization routine. The resulting minimum potential is returned and the values of the input parameters are set to the values that give a minimum. For example the tree-level effective potential of the electroweak model is given in figure 1.3. Passing the VEV parameter v into this routine returns -1.69×10^8 GeV^4 for the potential and sets the value of v to 246.0 GeV.

The one-loop potential can also be minimized in Effective. This uses the last calculation of the mass matrix (whether it is tree-level or one-loop) to compute the one-loop correction given in (6.38). This can then be minimized in the same manner as the tree-level potential.

Figure 6.7 shows a plot of the MSSM potential from Effective with the given parameter set $\{\mu = 30.0 \text{ GeV}, m_{H_1}^2 = 100.0 \text{ GeV}^2, m_{H_2}^2 = -609.0 \text{ GeV}^2, b = 7000.0 \text{ GeV}^2\}$ and the two gauge group couplings are set from the SM. In this plot we set $v_1^2 + v_2^2 = v^2 = (246.0 \text{ GeV})^2$ and $\tan\beta = \frac{v_2}{v_1}$. The tree-level effective potential of this model is derived

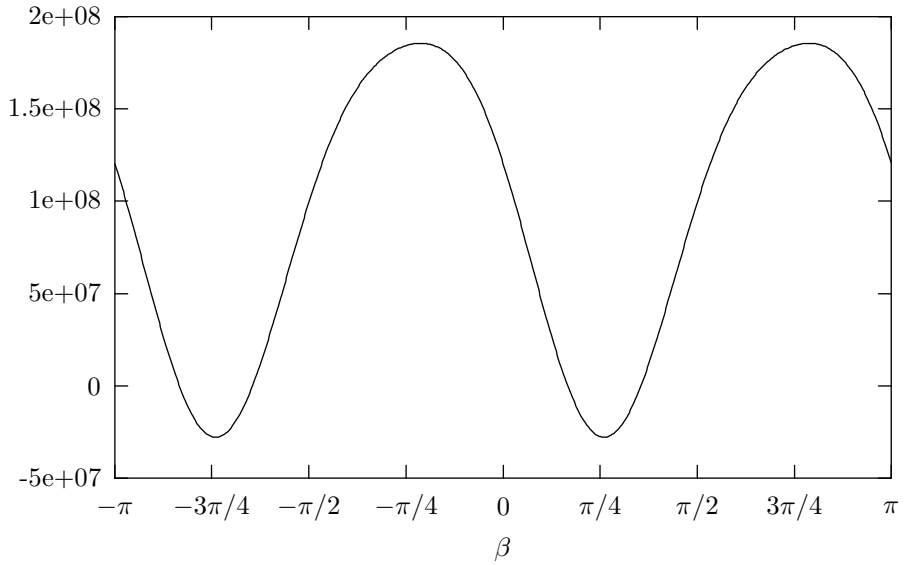


Figure 6.7: The effective potential of the MSSM for the parameter set $\{\mu = 30.0\ \text{GeV}, m_{H_1}^2 = 100.0\ \text{GeV}^2, m_{H_2}^2 = -609.0\ \text{GeV}^2, b = 7000.0\ \text{GeV}^2\}$ plotted in terms of β. This potential can be minimized in Effective over the parameters, v_1 and v_2.

in Effective and is given by

$$V_{\text{eff}} = \frac{1}{32}\left(g'^2 + g_W^2\right)\left(v_1^4 + v_2^4\right) - \frac{v_1 v_2 b}{2} + \frac{v_1^2 m_{H^1}^2}{2} + \frac{v_2^2 m_{H^2}^2}{2} + \frac{v_1^2 \mu^2}{2} + \frac{v_2^2 \mu^2}{2}. \quad (6.90)$$

From this equation we can see with the right set of parameters, the b term will push the potential below zero and require a particular combination of the VEVs to minimize.

6.5.4 Renormalization Group Equations

The last feature of a model that is generated automatically by Effective is the RGEs. Each parameter of the model can have its default value set at its own scale. The RGEs then evolve these parameters so that the values are all used consistently at any desired scale.

As was shown earlier the β functions for each parameter are generated using the one-loop effective potential and the two-, three- and four-point Green's functions and

the anomalous dimension of the external fields.

The solution to the boundary value problem can be given as follows. Start with a scale $x = M_x$. The boundary conditions at this scale can be given by

$$f(p(x)) = 0. \tag{6.91}$$

where p is the set of parameters which are defined at scale x. The boundary condition at a new scale, z, is

$$g(q(z)) = 0, \tag{6.92}$$

and again, q is the set of parameters defined at scale z. The full set of parameters of the model are given by $\{p(t), q(t)\}$ where $p(t)$ is constrained at $t = x$ and $q(t)$ is constrained at $t = z$. The parameters at the two scales are related by two functions, R^p and R^q by

$$\begin{aligned} q(z) &= R^q(q(x), p(x)) \\ p(z) &= R^p(q(x), p(x)), \end{aligned} \tag{6.93}$$

where these functions are determined by the RGEs. Using these equations we can numerically solve for the different parameters.

There are three numerical methods that have been considered in Effective [114]. The first is known as the "shooting" method [11]. Using this method one computes $g(q(z))$ numerically by use of the function R^q. This amounts to numerically finding the solution to

$$g(R^q(q(x), p(x))) = 0, \tag{6.94}$$

where $p(x)$ is the boundary values of the parameters at scale x. This just requires solving this equation over the set of parameters $q(x)$.

The "shooting" method cannot be used in SUSY, however, as SUSY requires the

input to be given at three scales: M_Z, M_S and M_x. Another reason that this method cannot be used for SUSY theories is that the boundary conditions of a SUSY theory cannot be conveniently expressed as is done in (6.92).

An alternative method [11] is the "drift" method. This method is a modified version of the "shooting" method. Here a guess $q_x = q(x)$ is taken and $p(x)$ is determined from (6.91). $q(z)$ and $p(z)$ are then given from (6.93). The boundary conditions at scale z are imposed to give $q(z)$. These are then run back up to the scale x by the inverse of (6.93). This gives a new set of values, $q'(x)$ and $p'(x)$. The conditions at scale x are then imposed again. This process defines a recurrence relation $q'(x) = Q(q_x)$ of which the desired values of p and q will be a fixed point. The problem with this method is that the physical values might not be the only fixed point and there is no guarantee that this fixed point is stable.

A third method, which has been implemented in Effective, handles the case where the fixed point is not stable. In this case we can instead numerically solve the equation

$$Q(q_x) - q_x = 0, \tag{6.95}$$

using the Newton-Raphson [11] method. This method can be rather slow but if one makes reasonable guesses for some of the less important parameters the parameter space can be reduced and convergence is quicker.

We have already seen the running of the gauge group couplings in fig. 6.5 produced by Effective. These were generated by setting the fields and groups of the model. Effective then computes the contribution to the β function from each field of the model. The coupling is evolved according to the β function generated.

Figure 6.8 shows the evolution of the b mass given by the β function for the Yukawa coupling in the SM. This has been evolved using β functions produced by Effective and

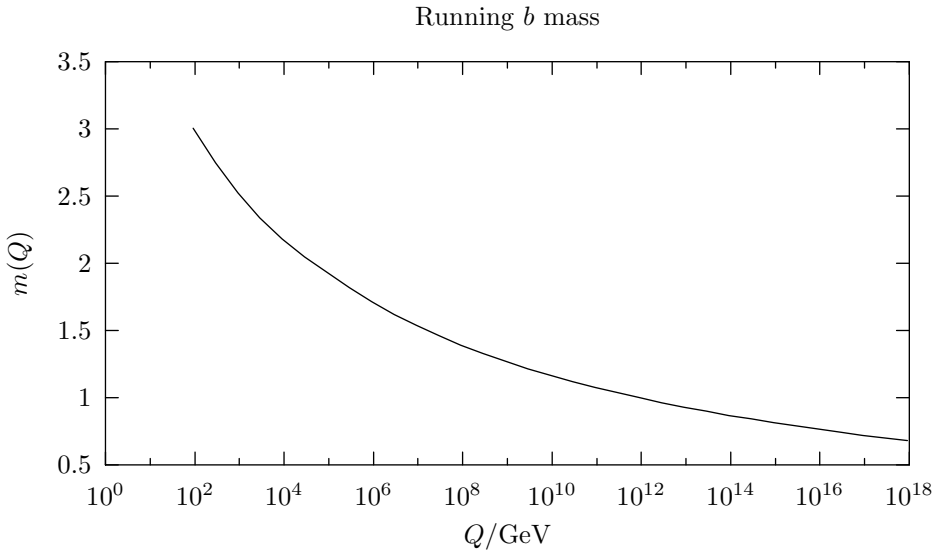


Figure 6.8: The running of the b mass. This is given by the β function of the Yukawa coupling.

the Runge-Kutta [11] method of differential equation evolution. The initial condition for this figure is $m_b(m_Z) = 3.0$ GeV. Again, the β function is generated by Effective. Each of the anomalous dimensions of the Yukawa coupling are added together. This is the sum from (6.81). The B factors have been calculated from the 1PI contributions of the vertices given in fig. 6.6.

6.5.5 Future Extensions of Effective

Before we discuss the extensions of Effective we first summarize what it can already do and how a user could use Effective without any extensions.

Effective is able to generate the full Lagrangian of a SUSY model based on the groups, fields, superpotential and SUSY breaking terms. This Lagrangian is used to generate the effective potential, mass matrices and RGEs of the model. At this stage a user can input their parameters over a range of scales and evolve them to the same scale. The mass spectrum at tree-level or one-loop can be evaluated given the parameter set. The user could then automate the study of the mass spectrum over a range of parameters.

The code of Effective allows for several extensions. The most useful one in producing physical results is allowing additional diagrams to be coded. The library provides a class `Diagram` which provides several useful functions. This class allows a user to pass in the external fields of a diagram and a set for each of the internal propagators. The class provides functions which find the couplings of a set of fields given as either interaction states, mass states or combinations of the two. This is done by computing the coupling of mass states based on their mixing matrices. The couplings can then be used in the computation of a diagram by inheriting the `Diagram` class and implementing the virtual function `ex function()` function. The diagram is then evaluated by iterating over all the internal propagator fields and summing the result of each diagram by a call to the `ex evaluate()` function. Implementing ones own diagrams would allow Effective to be used to produce results of physical calculations (like cross-sections) over a range of parameters.

The implementation of a users own diagrams isn't the only possible extension. The classes `GaugeField` and `MatterField` implement the set of expressions given in (6.86, 6.87). If someone wanted to consider a model which needed other terms (such as $N \neq 1$ SUSY) they would need to inherit these classes and override the function `ex interaction()`. They could then add new terms that depend on the fields. Unfortunately, this isn't all that would be needed for $N \neq 1$ SUSY. There would be a need to inherit the `Model` class and change several functions. But this is still a feasible task.

One may also consider extra dimensional models. These too could be implemented in Effective. One would have to implement a mechanism for producing a four dimensional effective Lagrangian from the input (groups, fields, superpotential, plus any extra interactions) but this too would be a feasible task. It may be that an implementation such as this could only be done for a specific type of compactification scheme. But would still be useful for studying both SUSY and non-SUSY models derived from an effective four

dimensional Lagrangian.

Another extremely useful extension that would be desirable is implementing a method of deriving the Feynman rules from the Lagrangian. All of the elements are in place to do such a task but would require quite a lot of work. These rules could be output in a format suitable for reading into an matrix element generator, like FormCalc, or one could even implement a matrix element generator as another extension of Effective. This would be a very complicated project, however.

Effective in its current form can mainly be used to study the mass spectrum of a model once the VEVs that minimize the effective potential are determined. This mass spectrum depends on the input parameter set which can be given at several scales and the RGEs of the model can be used to match them at the scale the mass spectrum is desired at. Future extensions of the software would allow this mass spectrum to be used in calculations of cross-sections and other properties. More substantial extensions could allow the software to be used on $N \neq 1$ SUSY and extra dimensional models. Finally the development of Feynman rules for a model could be implemented with an eye towards use in matrix element generators.

Conclusion

This thesis has presented two new software packages, Herwig++ and Effective. Herwig++ is a Monte Carlo event generator. In its current state, this generator is able to generate e^+e^--annihilation events. Chapter 2 has developed a new set of variables for the parton shower and chapter 3 has presented an improved hadronization model. A full discussion of the implementation specific features of Herwig++ have been given in chapter 4. Results of the new models and the package as a whole have been shown in chapter 5. These results give us great confidence that the new shower and hadronization are a better description of the physics.

Currently, we are working on Herwig++ to generate initial-state showers. This will enable us to describe hadron-hadron and lepton-hadron events. The aim is to have this fully functional for use with Tevatron and LHC studies. Once the initial-state shower is fully functional a new underlying event model can be implemented as well as improving the hadronic decays. Work is already underway to include spin correlations into the shower and decays. New matrix elements for various processes can be implemented and SUSY particles and processes can be included. As the history of HERWIG has shown, there will continue to be a lot of potential for future research in Herwig++.

Effective is a model building program. This program has been designed to work with $N = 1$ SUSY models but future extensions could move it beyond just these models. This program automates the process of determining the mass spectrum of a model for a given

parameter set. The mass spectrum generated with Effective can be read into Herwig++ to define the masses of the supersymmetric particles in Herwig++. This would provide an automated way of studying SUSY at colliders over a range of parameters. Currently, only special purpose programs, like SOFTSUSY [116], are able to do something similar. Future extensions of Effective may provide more functionality and make it an extremely powerful and useful tool for studying models.

Appendix A

Herwig++

This appendix contains a few relevant issues pertaining to the use and future development of Herwig++. I first give an example analysis program to count the number of π^0 in an event. Following that is a brief discussion of how to change the parameters in Herwig++. Finally, I give the skeleton structure of how one would implement one's own matrix element.

A.1 Counting Pions

This section provides a description of the functions needed to run a simple Herwig++ program. The program given here will simply count the number of neutral pions in the final state.

Herwig++ can be run inside an exception handling clause. In C++ this is the statement

```
try { ...your code here ...}
catch(...)  { // do something with the exception }
```

Herwig++ and ThePEG will throw exceptions to explain why the program has terminated. If these aren't caught the message associated with the end of the program won't be known and the reason for the termination will remain a mystery. Therefore, the main program to run Herwig++ will be enclosed in such a statement and the message from the exception will be printed. The exact form we use is

```
try { ...generate events/analyze ...}
catch(std::exception & e) {
   cerr << e.what() << endl;
   return 1;
} catch(...)  {
   cerr << "Unknown exception\n";
   return 2;
}
```

We now move on to discussing what is put in the "generate events/analyze" section. A helper class `HerwigRun` has been developed for Herwig++. This class takes the command line arguments (`int argc, char **argv`) and sets up Herwig++ to either initialize, read or run. These different stages are discussed in the next section. The command to create a `HerwigRun` class is

```
Herwig::HerwigRun hw(argc,argv);
```

Once this class is created generating an event is straightforward. We first must check the status of the program, given by the functions `isRunMode()` and `preparedToRun()`. We can then iterate over the number of events to generate given by the function `getN()`. To generate an event we simply call `generateEvent()`. The code to do all of this is

```
if (hw.isRunMode() && hw.preparedToRun()) {
   for(int i = 0; i<hw.getN(); i++) {
      hw.generateEvent();
```

At this point we have now generated an event. This is where the analysis code must be implemented. The `HerwigRun` class also offers a function to retrieve the particles from the last event generated. `ThePEG::tPVector getFinalState(int i)` is this function. If `i` is not given (e.g. `getFinalState()`) then the final state particles of the event are returned. If the `i` argument is given this function returns the particles in the final state given by the step `i`.

Returning to our example of counting the number of π^0 particles in the final state we pass the vector of final state particles to a counting function which we define, `int countPions(ThePEG::tPVector particles)`. This function is fairly straightforward and is given by

```
int countPions(ThePEG::tPVector particles) {
   ThePEG::tPVector::iterator it;
   int count = 0;
   for(it=particles.begin(); it!=particles.end(); it++) {
      if((*it)->id() == ThePEG::ParticleID::pi0) count++;
   }
   return count;
}
```

This function has relied on the class `ThePEG::Particle` which defines several functions of the particle. Here we have used the `long id()` function to determine the PDG code of the particle. There are many more functions that would be useful for analysis. The user is encouraged to read the documentation of the `Particle` class for more details.

The `HerwigRun` class also defines functions which return some of the elements from `ThePEG`. These are things like the `Step` or the `CollisionHandler` from which the user is able to obtain all the information of the event.

A.2 Repository

The repository is a feature of ThePEG that allows the parameters of the program to be changed without having to recompile the code. These parameter files can then be exchanged within collaborations to ensure that researchers are working with the same parameters.

The repository can be thought of as the input files to Herwig++. These are written in a human readable form, though the repository has a simple syntax of its own. The input files can then be saved in machine readable form for faster access on subsequent runs. In Herwig++ there is a default input file, `HerwigDefaults.in`. This sets up all the relevant objects and instructs the program to allocate memory for the various parts of the event generator. This initial setup can be saved in a machine readable file, `HerwigDefaults.rpo`. The `HerwigRun` class reads in the command line argument `init`. This instructs the class to create the `.rpo` from the `.in` file. `HerwigDefaults.in` also reads `Shower.in`, `Hadronization.in` and `Decays.in` as well as the particles in `ThePEGParticles.in` and `HerwigParticles.in` and the decay modes from `HwDecays.in`.

The `HerwigDefaults.in` currently defines two types of generators. One for LHC-like events and another for LEP-like events. These two generators can then have their parameters changed (such as c.m. energy or various cuts) by editing the files `LHC.in` or `LEP.in`, respectively. These modifications are read in by the `read` command line argument. This reads in the appropriate `.in` file and produces a `.run` file. The `.run` is checked to ensure that all the relevant objects and links have been set so that the event generator can run.

The last stage is to actually run some events. Using the command line argument `run` reads in the appropriate `.run` file and generates events. There is a parameter

`NumberOfEvents` in each generator object that sets the maximum number of events to run. The `HerwigRun` class provides for a new number of events to be given at runtime. This number must be less than or equal to the parameter given to the `NumberOfEvents` field. To instruct the program to use a new number simply use the command line argument `-N #` when starting the program.

To summarize there are three ways in which a program that uses a `HerwigRun` class can by used. These are given as

```
Program_name init
Program_name read Generator_file.in
Program_name run Generator_file.run -N #
```

There are more commands that can be given in the command line but these mostly change some of the lower level instructions. The full set of commands is

```
Program_name init|read|run [-N num-events] [-seed random-generator-seed]
[-d debug-level] [-dHw herwig-debug-level] [-l load-path] [-L first-load-path]
[-r repo-file] [-i initialization file] [run-file]
```

To change the value of a parameter you find the line in the input file that this is governed by. For example if we wanted to turn the initial-state radiation off we would look in the `Shower.in` file. This file has the line

```
set theSplittingGenerator:OnOffISRMode 1
```

If we simply edit this file and change the 1 into a 0 this will turn the initial-state radiation off. Most of the commands are straightforward like this. There are other commands in the repository than `set`. A relevant subset is `create,set,mkdir,cd,library`. The `create` command is used to define a new object given by a class. The `set` command is used to change one of the parameters of that object. The repository is structured like a file system. You can create the objects you want in directories in order to keep things ordered. The `mkdir` command creates a new directory and the `cd` command changes

to the given directory. The last command is the `library` command. This is used to dynamically load a library. This must be done to define objects that are in a library that hasn't been loaded yet. For example, if we want to define an `AmegicInterface` object we would have to load the `libHwAmegic.so` library first.

A.3 Matrix Element Development

In this last section I show how a user would implement their own matrix element. The class `MEBase` defines the lowest level of matrix element abstraction. To implement a matrix element one must at least inherit this class and define the functions

```
unsigned int orderInAlphaS() const
unsigned int orderInAlphaEW() const
double me2() const
Energy2 scale() const
generateKinematics(const double r)
CrossSection dSigHatDR() const
void getDiagrams() const
Selector<const ColourLines*> colourGeometries(tcDiagPtr diag) const
```

This list looks quite daunting. ThePEG has already generated these functions for special types of matrix elements. For example the class `ME2to2Base` as able to define the `generateKinematics()`, `dSigHatDR()` as well as the `scale()` for all $2 \to 2$ processes. A further subclass `ME2to2QCD` defines `orderInAlphaS()` and `orderInAlphaEW()` functions as well as providing routines for general functions like determining the number of active flavours at a given scale.

If we take an example to inherit the `ME2to2QCD` class this means we only need to define the functions

```
double me2() const

void getDiagrams() const

Selector<const ColourLines*> colourGeometries(tcDiagPtr diag) const
```

The function `double me2() const` is self explanatory. This is just the value of the matrix element for the set of currently generated points. To access the set of points there is a function `meMomenta()` which returns a vector of `Lorentz5Momentum` objects. This vector contains the momentum generated for all the incoming and outgoing particles.

The `getDiagrams() const` class is used to indicate to the matrix element what are the diagrams used to produce the matrix element (squared). There is a function `add(DiagPtr)` which adds a diagram to the list. There is a helper method defined to define a diagram. For example

```
new_ptr((Tree2toNDiagram(2), q, qb, 1, gamma, 3, l, 3, lb, -1))
```

define a $2 \to 2$ process where the `q` and the `qb` are the incoming particles. The 1 indicates that the `gamma` particle is the child of the first incoming particle. The `3,l,3,lb` series indicates that both the `l` and `lb` particles are the children of the `gamma` particle. The last argument of -1 indicates that this is the end of the diagram definition. This is a flag that must be negative but can have different values. The different values are used to determine the different types of diagrams. If one wanted to weight the different diagrams they would need to overload the `Selector< DiagramIndex> diagrams(const DiagramVector &) const` function. In this function one can iterate over the diagrams and add them into the `Selector` object with a weight. When a diagram is selected the weights are taken into account.

The last function that must be implemented is the `Selector<const ColourLines *> colourGeometries(tcDiagPtr diag) const` function. This function allows you to set the different colour connections. These can also be given a weight which the particular colour connection implemented is chosen from.

Having implemented all of these functions the matrix element is defined and can be used in Herwig++. The typical place to use the matrix element is in the hard subprocess. It is possible to use a matrix element in other processes, such as in the decays, but that requires more development and working with the code on a lower level.

Appendix B

Effective

This appendix is devoted to coding issues of Effective. I give here an example of coding the MSSM model in Effective and how one can use the model to get the mass of the neutralinos. I also show how to evolve the RGEs to provide the scale dependence of the parameters. At the end I give a skeleton outline of how one would implement one's own groups and diagrams.

B.1 MSSM

The main driver of Effective is the class `Model`. This class is what directs all the calculations and provides all the components. This class that must be inherited to implement ones own model. The methods that need to be defined are

```
void createGaugeGroups()
void createGaugeFields()
void createMatterFields()
void addOtherTerms()
ex superPotential()
```

The `createGaugeGroups()` function specifies what groups are used in the model. For example in MSSM and the SM we have only the $SU(3)_c$, $SU(2)_L$ and $U(1)_Y$ groups. These can be implemented via the function

```
void MSSM::createGaugeGroups() {
  addGaugeGroup(new U1Group("U1", "{g'}", this, U1Y));
  addGaugeGroup(new SU2Group("SU2", "{g_W}", this, SU2w));
  addGaugeGroup(new SU3Group("SU3", "{g_3}", this, SU3c));
}
```

The objects `U1Y`, `SU2w` and `SU3c` are all integers that define the 'line' of the group. The two strings for each group define the name of the group coupling in the standard text output and the LaTeX output, respectively. The groups can then be accessed at a later point using the standard text string (e.g. `getGroup("U1")`).

The next function that needs to be implemented is `createGaugeFields()`. This function defines the fields that mediate the interactions. In the MSSM these are the superpartners $(B^\mu, \tilde{B})$, $(W_i^\mu, \tilde{W}_i)$ and $(A_a^\mu, \tilde{A}_a)$. These are given by

```
void MSSM::createGaugeFields() {
  addField("B",new GaugeField("B","B",Utils::LorentzVector,
                              getGaugeGroup("U1")));
  addField("Bino",new GaugeField("Bino","\\tilde{B}",Utils::WeylSpinor,
                              getGaugeGroup("U1"),getGaugeField("B")));
  addField("W",new GaugeField("W","W",Utils::LorentzVector,
                              getGaugeGroup("SU2")));
  addField("Wino",new GaugeField("Wino","\\tilde{W}",Utils::WeylSpinor,
                              getGaugeGroup("SU2"),getGaugeField("W")));
  addField("gluon",new GaugeField("A","A",Utils::LorentzVector,
```

```
                                getGaugeGroup("SU3")));
  addField("gluino",new GaugeField("Aino","\\tilde{A}",Utils::WeylSpinor,
                                getGaugeGroup("SU3"),getGaugeField("gluon")));
}
```

The `addField()` function takes a string as an index for the field and a pointer to a field. This same function is used to add the `MatterField`s. Creating a `GaugeField` requires two strings. Again the first is the text output string and the second is the LaTeX output. These strings are followed by the spin of the particle. This is actually the $(2s+1)(-1)^{2s}$ factor. This has been conveniently defined in the class `Utils` for the scalar, fermion and vector spin types. The spin is then followed by a pointer to the gauge group that the field is the mediator of. The last argument, which is optional, is the SUSY partner of the field. We can see above that the SUSY partner is set for the fermion but not the vector. This is because one field must be created first. Once created the partner can be set. Setting it in the creation of the fermion also sets it in the vector, so this syntax sets both partners to each other with only one reference to a partner.

After the `addGaugeFields` function is defined we implement the `addMatterFields` function. This function sets all of the remaining fields in the model. Again looking at our example from MSSM we can write the function as

```
void MSSM::createMatterFields() {
  numeric half(1,2);
  numeric sixth(1,6);
  numeric third(1,3);
  numeric twothirds(2,3);
  // Adding leptons/sleptons
  addField("leptonL",new MatterField("L","\\ell",Utils::WeylSpinor,famsize,
                                     getGaugeGroup("U1"),-half,
```

```
                                    getGaugeGroup("SU2"),1));
addField("sleptonL",new MatterField("sL","\\tilde{\\ell}",
                                     Utils::Scalar,famsize,
                                     getGaugeGroup("U1"),-half,
                                     getGaugeGroup("SU2"),1,
                                     getMatterField("leptonL")));

addField("leptonR",new MatterField("eR","e_R",Utils::WeylSpinor,famsize,
                                    getGaugeGroup("U1"),-1));
addField("sleptonR",new MatterField("seR","\\tilde{e_R}",
                                     Utils::Scalar,famsize,
                                     getGaugeGroup("U1"),-1,
                                     getMatterField("leptonR")));

// Adding quarks and squarks
addField("quarkL",new MatterField("Q","Q",Utils::WeylSpinor,famsize,
                                   getGaugeGroup("U1"),sixth,
                                   getGaugeGroup("SU2"),1,
                                   getGaugeGroup("SU3"),1));
addField("squarkL",new MatterField("sQ","\\tilde{Q}",
                                    Utils::Scalar,famsize,
                                    getGaugeGroup("U1"),sixth,
                                    getGaugeGroup("SU2"),1,
                                    getGaugeGroup("SU3"),1,
                                    getMatterField("quarkL")));

addField("uR",new MatterField("uR","u_R",Utils::WeylSpinor,famsize,
```

```
                                getGaugeGroup("U1"),twothirds,
                                getGaugeGroup("SU3"),1));
addField("suR",new MatterField("suR","\\tilde{u_R}",Utils::Scalar,famsize,
                                 getGaugeGroup("U1"),twothirds,
                                 getGaugeGroup("SU3"),1,
                                 getMatterField("uR")));

addField("dR",new MatterField("dR","d_R",Utils::WeylSpinor,famsize,
                                getGaugeGroup("U1"),-third,
                                getGaugeGroup("SU3"),1));
addField("sdR",new MatterField("sdR","\\tilde{d_R}",Utils::Scalar,famsize,
                                getGaugeGroup("U1"),-third,
                                getGaugeGroup("SU3"),1,
                                getMatterField("dR")));

// Higgs fields 1 is upper, 2 is lower
addField("H1", new MatterField("H1","{H^1}",Utils::Scalar,1,
                                 getGaugeGroup("U1"),-half,
                                 getGaugeGroup("SU2"),1));
addField("sH1", new MatterField("sH1","\\tilde{H^1}",Utils::WeylSpinor,1,
                                  getGaugeGroup("U1"),-half,
                                  getGaugeGroup("SU2"),1,
                                  getMatterField("H1")));

addField("H2", new MatterField("H2","{H^2}",Utils::Scalar,1,
                                 getGaugeGroup("U1"),half,
                                 getGaugeGroup("SU2"),1));
```

```
addField("sH2", new MatterField("H2","\\tilde{H^2}",Utils::WeylSpinor,1,
                                getGaugeGroup("U1"),half,
                                getGaugeGroup("SU2"),1,
                                getMatterField("H2")));
// Now add higgs Vev
Parameter upsilon1 = addParameter("HiggsVev1","\\upsilon_1",220.0,
                                  Parameter::vev);
Parameter upsilon2 = addParameter("HiggsVev2","\\upsilon_2",220.0,
                                  Parameter::vev);
addVev("HiggsVev1",getField("H1"), lst(getIndex("H1","SU2")==1),
       (upsilon1+Model::star));
addVev("HiggsVev2",getField("H2"), lst(getIndex("H2","SU2")==2),
       (upsilon2+Model::star));
addVevParameter(upsilon1);
addVevParameter(upsilon2);
```

The meaning of this code is easy to see. Again we see the last (optional) argument in creating a `MatterField` is the superpartner. We can see now that there are some new arguments to these fields though. There is a `famsize` argument. This is an integer which defines how many generations the field has. For example in MSSM we would set `famsize` to 3. Lastly we see that after each gauge group is a number. This is the 'charge' of the group. Most groups just have a charge of 1. The $U(1)_Y$ does not, however. This has fractional charges. The class `numeric` is one from GiNaC which defines rational numbers, without rounding errors. These expressions will be printed in LaTeX as they are (e.g. `half` $= \frac{1}{2}$).

In the last part of the code we see that we have created the VEVs. The class `Parameter` is used in Effective to represent all of the parameters of the model. The

`Parameter` class is printed out in symbolic form during printing but is evaluated to its value during calls to `evalf()`. The object `Model::star` is a placeholder which is where the relevant field is placed. If we wanted to also add an imaginary VEV would could use a similar syntax `imagVEV+Model::star` and in this instance `Model::star` is the imaginary part of the field.

As discussed in the text the parameters could be added in a different way. If we wanted to work with v and $\tan\beta$ rather than v_1 and v_2 we could have used the code

```
Parameter upsilon = addParameter("HiggsVev","$\backslash\backslash$upsilon",246.0,
                                 Parameter::vev);
Parameter beta = addParameter("beta","$\backslash\backslash$beta",1.1,
                              Parameter::vev);
addVev("HiggsVev1",getField("H1"), lst(getIndex("H1","SU2")==1),
       (upsilon*sin(beta)+Model::star));
addVev("HiggsVev2",getField("H2"), lst(getIndex("H2","SU2")==2),
       (upsilon*cos(beta)+Model::star));
addVevParameter(upsilon);
addVevParameter(beta);
```

but as discussed in the text, this type of setup will not properly define the RGEs and the minimization routine may not succeed.

In the MSSM there are many terms which must be added from both the superpotential and from the soft breaking terms. We won't give them all here but instead will present an example of how to add a term. We will look at how to add the slepton mass term. This is given by

```
ex m0LL = addFamilyMatrix("m0LL", "{m^2_L}", famsize);
vector<idx*> sumIndex;
```

```
idx i = Utils::familyIndex(0,famsize);
idx j = Utils::familyIndex(1,famsize);
MatterField *sleptonL = getMatterField("sleptonL");
ex li = sleptonL->expression();
ex lj = Utils::conjugate(li);
lj = lj.subs(sleptonL->familyIndex()==j);
sumIndex = sleptonL->getIndices();
sumIndex.push_back(&j);
ex sleptonLTerm = Utils::real(Utils::sumIndices(m0LL*li*lj,sumIndex));
add(-sleptonLTerm);
```

At first this looks quite horrific. But once the different parts are explained it is obvious what everything means. The first term is used to define the slepton mass term. Since there are three slepton generations the mass-squared coupling must be defined as a 3×3 matrix of parameters. The `addFamilyMatrix()` function provides this. This function creates a matrix whose elements are parameters. The parameters are created with the indices appended to the name (e.g. the m^2_{L11} element is given by the parameter `m0LL11`). When the matrix is created it is created as an identity matrix.

The next line defines a vector in which all the indices which are summed over are placed. This is followed by defining two family indices. By default the `FamilyMatrix` class uses the first two indices from the `Utils::familyIndex()` function. The `Utils` class defines some universal symbols for different indices. A call to the `familyIndex` function returns one of the symbols (given by the first integer) as an index where the dimension of the index is given by the second argument (e.g. the number of families). As a note, there is a integer `Utils::max_indices` which defines how many symbolic indices are available.

The definition of the two family indices, `i` and `j`, is followed by a retrieval of the

`MatterField` pointer to the slepton. If we refer back to our definition of the slepton field we can see that the first string is "slepton". This is the tag that is used to retrieve the field at a later time. There is also a function `getGaugeField` which behaves in a similar manner, only it returns a `GaugeField` pointer. After the retrieval of the slepton pointer we now get the expression of the slepton. As Effective seperates the real and imaginary parts of the scalars and fermions, the expression is $\frac{1}{\sqrt{2}}(\mathcal{R}(f)+i\mathcal{I}(f))$. Also the function expression returns the fields with each index attached. For example a left-handed quark would have its family index as well as the $SU(2)_L$ and $SU(3)_c$ indices added. A vector also has a `LorentzIndex` attached. The function `Utils::conjugate` then replaces all i's with $-i$. The default family index of a field is taken from `Utils::familyIndex(0,famsize)`. Since we want to contract the family indices of the mass-squared coupling we must replace this index with `Utils::familyIndex(1,famsize)` in the conjugate expression. This is done via the `subs()` command. This command takes two forms. The first is to use the `==` sign. This sets the argument on the left to be replaced by the argument on the right. This is useful when making one substitution. When many substitutions are desired we can instead use the form `subs(lst,lst)`. The first `lst` is replaced by the second. A `lst` is a class from GiNaC and is just a sequence of expressions.

After the expression for ϕ and ϕ^* have been retrieved we want to contract all the indices and create the expression $m^2_{Lij}\phi^*_j\phi_i$. To do this we store all of the indices given in the slepton into the `sumIndices` vector we created earlier. We also realize that this vector doesn't contain the extra family index in m^2_L, and now in `lj`, so we also add the index to the back of the list. Once the `sumIndices` vector contains all the indices which we are summing over we can call the routine `Utils::sumIndices(ex,vector<idx*>)`. This routine iterates the list of indices over all permutations and evaluates the ex at each set of index values. The sum of all these permutations and evaluations is returned. The last command on this line is the `Utils::real`. This returns all the parts of the expression that are not multiplied by i. In this example we shouldn't need to call this

command as $\phi^*\phi$ should be a real quantity. It is also worth noting that instead of coding an expression as exp + c.c. it is easier and faster to write `2*Utils::real()`.

In the end we call the `add()` function. This adds the expression to the Lagrangian. If we were developing terms of the superpotential we would instead sum all the contributions from all the terms and return it. We don't want to call the `add()` routine with the superpotential.

B.2 Plotting Effective Potential and Running Couplings of MSSM

Once we have developed the class `MSSM` with all the relevant groups, fields and terms, we can work with this program. The following code is used to create the model and initialize it

```
MSSM mssm;
Model::readCommandLine(argc,argv,&mssm);
mssm.couplings("mssm.gar");
mssm.initialize();
```

The first line creates an `MSSM` object. The second line calls a static function from the `Model` class which reads in standard command line arguments and sets the relevant parameters of the model. The possible arguments are

```
-p # - Sets the evaluation of the effective potential to tree-level (0)
       or one-loop (1)
-t # - Sets the evaluation of the tadpoles to tree-level (0) or
       one-loop (1)
```

```
-a # - sets the evaluation of the mass matrices to tree-level (0) or
       one-loop (1)
-r   - Rederives the couplings from scratch
-s   - Saves the couplings to the file provided
-h   - Prints a help message
```

The third line of code provides the model with a file to read the couplings in from and save to, depending on what the program has been specified to do. The last line initializes the model. This means generate (or read) the couplings, find all the tree level mass matrices and generate the RGEs. At this stage the model is able to be used.

At this point we will want to read our parameters in from a file. This is done by the `loadParameters()` function. The code that would read the parameters from the file `MSSM.dat` is

```
ifstream parin;
parin.open("MSSM.dat");
mssm.loadParameters(parin);
parin.close();
```

This file has a simple structure. It is tab delimited where the first entry is the name of the parameter (the text name given during its definition) and the value of the parameter. There is also a parameter defined for every model. This is `renormScale` and defines the current renormalization scale. It is defaulted to 91.2 GeV (the Z^0 mass) and when a parameter is read in it is assumed to be entered at the scale currently given by `renormScale`. So if one wanted to input parameter `a` at 91.2 GeV but parameter `b` at 200 GeV they would use

```
a 50.0
```

```
renormScale 200.0
b 20.0
```

Once the parameters have been read in one may want to evolve the RGEs to a scale. At first the boundary conditions must be determined at one scale. This is done by calling `mssm.rges().matchScales()`. This routine goes through the boundary conditions entered and uses the Newton-Raphson method to find a suitable set of parameters that match all the boundary conditions given. Once that is done the parameters can be evolved to the scale desired by calling `mssm.rges().evolve(scale)`.

Once the parameters have been determined at the scale desired the mass spectrum can be developed. This is done with a call to `mssm.masses().diagonalize()`. This will diagonalize all the mass matrices at either tree-level or one-loop. There is an additional flag, `Model::CorrectMassStates` which determines whether the one-loop corrections are applied directly to the mass eigenstates derived from the tree-level mass matrices, or if the one-loop corrections are applied based on the interaction eigenstates. If one wanted to get the mass of a particle they simply call `mssm.masses().massOf(field)` where field is the expression with all the indices evaluated (e.g. instead of looking at the top quark mass, one would have to look at the left-handed quark, with $SU(2)_L$ index = 1, family index = 3 and the $SU(3)_c$ index set to any number between 1 and 3). In the SM, for example, one would fully expect to find that the mass all of the colour indices of a particular quark are the same. But it would be possible to develop a theory where this is not the case.

The last function that would be desired is to find the effective potential. This can be given as an analytic expression by a call to `mssm.treePotential()` or at one-loop order by a call to `mssm.potential()` (with the correct value of `Model::_potApprox` set). Using this we can use the code

```
ofstream potPlot;
potPlot.open("MSSM_Pot.dat");
double v1, v2;
ex pot = mssm.treePotential();
double v = 246.0;
double beta;

for(beta = -3.14; beta < 3.14; beta += 0.01) {
  mssm.getParameter("HiggsVev1") = v*cos(beta);
  mssm.getParameter("HiggsVev2") = v*sin(beta);
  potPlot << beta << "\t" << pot.evalf() << endl;
}
potPlot.close();
```

to generate fig. 6.7.

B.3 Entering a New Group

When studying models different symmetries are desired. These are given by the group governing the interaction. Effective provides a way in which a new group can be defined. Inheriting the class `GaugeGroup` is how this is done. This class requires the user to implement

```
constructor(string n, string cn, Model* m, char l);
ex structureConstant(GaugeIndex&, GaugeIndex&, GaugeIndex&) const;
ex generator(GaugeIndex&, MatterIndex&, MatterIndex&) const;
GaugeIndex gaugeIndex(int i=0) const;
```

```
MatterIndex matterIndex(int i=0) const;

bool isIndex() const;

ex Cr(bool) const;

ex C2(bool) const;
```

The constructor defines the coupling (given by the text name `n` and the LaTeX name `cn`). The `char l` is used to define the line of the group. A constructor is generally of the form `ClassName(n,cn,m,l) : GaugeGroup(n,cn,m,l){}`. This just passes all the handling of these arguments to the `GaugeGroup` superclass. The functions `C2(bool)` and `Cr(bool)` are used to give the evalation of $t_r^a t_r^a = C_2(r) \cdot \mathbf{1}$ and $C(r) = \mathrm{Tr}[t_r^a t_r^a]$. If the boolean passed in is true then the adjoint representation is used. If it is false then the fundamental representation is used.

Next one must define the indices. This usually amounts to defining a set of symbols for the adjoint and fundamental indices. The indices are then just `idx` classes with the symbol and the correct dimension set. The last remaining pieces are the structure constants and generators, given by `structureConstant` and `generator`. These require the definition of a `tensor` class from GiNaC. An example of an $SU(3)$ structure constant is given here. The definition of the tensor is

```
class SU3Structure : public tensor {
  GINAC_DECLARE_REGISTERED_CLASS(SU3Structure,tensor)

 public:
  void print(const print_context &c, unsigned level = 0) const;
  ex eval_indexed(const basic &i) const;
};
```

One can then define a helper function `SU3_structure` which returns the expression for the structure constant given the indices. This takes the form

```
inline ex SU3_structure(const ex &i1, const ex &i2, const ex &i3) {
  if(!is_a<GaugeIndex>(i1) || !is_a<GaugeIndex>(i2) ||
     !is_a<GaugeIndex>(i3))
    throw(std::invalid_argument(
          "Indices to SU(3) structure must be of type GaugeIndex"));

  if(ex_to<GaugeIndex>(i2).line() !=
     ex_to<GaugeIndex>(i3).line())
    throw(std::invalid_argument(
          "GaugeIndices must be of the same line and type"));

  return indexed(SU3Structure(), sy_anti(), i1, i2, i3);
}
```

The structure constant can then be defined by

```
GINAC_IMPLEMENT_REGISTERED_CLASS(SU3Structure,tensor)
DEFAULT_CTORS(SU3Structure)
DEFAULT_ARCHIVING(SU3Structure)
DEFAULT_COMPARE(SU3Structure)
```

and the `print` and `eval_indexed` functions are

```
void SU3Structure::print(const print_context & c, unsigned level) const {
  if (is_a<print_tree>(c)) inherited::print(c, level);
  else {
```

```
    ostream& os = c.s;
    os << "f";
  }
}

ex SU3Structure::eval_indexed(const basic &i) const {
  GINAC_ASSERT(is_a<indexed>(i));
  GINAC_ASSERT(i.nops() == 4);
  GINAC_ASSERT(is_a<SU3Structure>(i.op(0)));
  GINAC_ASSERT(is_a<GaugeIndex>(i.op(1)));
  GINAC_ASSERT(is_a<GaugeIndex>(i.op(2)));
  GINAC_ASSERT(is_a<GaugeIndex>(i.op(3)));

  const GaugeIndex& g1 = ex_to<GaugeIndex>(i.op(1));
  const GaugeIndex& g2 = ex_to<GaugeIndex>(i.op(2));
  const GaugeIndex& g3 = ex_to<GaugeIndex>(i.op(3));

  // Numeric Evaluation
  if(static_cast<const indexed &>(i).all_index_values_are(info_flags::integer))
  {
    int i = ex_to<numeric>(g1.get_value()).to_int();
    int j = ex_to<numeric>(g2.get_value()).to_int();
    int k = ex_to<numeric>(g3.get_value()).to_int();
    int ip = i; int jp = j; int kp = k;
    if(ip>jp) Utils::swap(ip,jp);
    if(jp>kp) Utils::swap(jp,kp);
    if(ip>jp) Utils::swap(ip,jp);
```

```
    static ex half = numeric(1,2);
    double cyc = Utils::cyclicPermutation(i,j,k);
    if(ip==1) {
      if(jp==2 && kp==3) return cyc;
      else if(jp==4 && kp==7) return half*cyc;
      else if(jp==5 && kp==6) return -half*cyc;
    } else if(ip==2) {
      if(jp==4 && kp==6) return half*cyc;
      else if(jp==5 && kp==7) return half*cyc;
    } else if(ip==3) {
      if(jp==4 && kp==5) return half*cyc;
      else if(jp==6 && kp==7) return -half*cyc;
    } else if(ip==4 && jp==5 && kp==8) return sqrt(numeric(3,4))*cyc;
    else if(ip==6 && jp==7 && kp==8) return sqrt(numeric(3,4))*cyc;
    return 0;
  }
  // No further simplification
  return i.hold();
}
```

This then completely defines the behaviour of the structure constants of the group. A similar thing can be done for the generators.

B.4 Defining a Diagram

Defining a diagram is a useful way to do computations with Effective. This is also straightforward. One simply inherits the `Diagram` class and implements the constructor

```
MyNewDiagram(DElementVec &v, Model *m) : Diagram(v,m) {}
```

and the `function()` function. This function is used to evaluate the diagram at the current set of fields. The `DElementVec` is used to specify a set of fields for each external leg and propagator of the diagram. When the diagram is created the desired choice of fields for each part is given. For example we want to compute the cross section for $e^+e^- \to q\bar{q}$ at tree-level. We simply create some `DiagramElements` and put the appropriate fields in.

```
DElementVec eeqq(4);
exvector electrons, quarks;
electrons.push_back(real_left_electron);
electrons.push_back(imag_left_electron);
electrons.push_back(real_right_electron);
...
eeqq[0] = DiagramElement(electrons, true);
eeqq[1] = DiagramElement(electrons, true);
eeqq[2] = DiagramElement(quarks, true);
eeqq[3] = DiagramElement(quarks, true);
```

This creates the `DElementVec` object with all the electrons and all the quarks. This then gets given as input to the diagram `ee2qq`

```
ee2qq myee2qq(eeqq,my_model);
cout << "My matrix element is " << myee2qq.evaluate() << endl;
```

The `evaluate` function goes over every permutation of the fields given in the four `DiagramElement`s and computes a contribution to the matrix element. This contribution is given by the `function` function. This must be defined so that each combination

of fields contributes the relevant part. This is done by determining the couplings of the set of fields given. For example

```
ex coup123 = coupling(0,1,2);
ex coup234 = coupling(1,2,3);
ex coup134 = coupling(0,2,3);
ex coup1234 = coupling(0,1,2,3);
```

The `true` statement in the definition of the `DiagramElement` indicates that the mass state couplings are to be used. If it were false the interaction couplings would be used.

Bibliography

[1] S. Weinberg, *The Quantum Theory of Fields*, vol. I. Cambridge University Press, Cambridge, UK, 1996.

[2] S. Weinberg, *The Quantum Theory of Fields*, vol. II. Cambridge University Press, Cambridge, UK, 1996.

[3] M. Peskin and D. Schroeder, *An Introduction to Quantum Field Theory*. Perseus Books, 1995.

[4] R. K. Ellis, W. J. Stirling and B. R. Webber, *QCD and Collider Physics*, vol. 8 of *Cambridge Monogr. Part. Phys. Nucl. Phys. Cosmol.* Cambridge University Press, 1 ed., 1996.

[5] J. Collins, *Renormalization*. Cambridge Monographs on Mathematical Physics. Cambridge University Press, 1986.

[6] L. H. Ryder, *Quantum Field Theory*. Cambridge University Press, 2nd ed., 1985.

[7] **Super-Kamiokande** Collaboration, K. Scholberg, *Atmospheric Neutrinos at Super-Kamiokande*, `hep-ex/9905016`.

[8] **D0** Collaboration, *A precision measurement of the mass of the top quark*, *Nature* **429** (2004) 638–642.

[9] D. Griffiths, *Introduction to Elementary Particle Physics*. John Wiley and Sons, 1987.

[10] F. Halzen and A. Martin, *Quarks and Leptons*. John Wiley and Sons, 1984.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling and B. P. Flannary, *Numerical Recipes in C*. Cambridge University Press, second ed., 1999.

[12] W. J. Stirling, R. Kleiss and S. D. Ellis, *A new Monte Carlo treatment of multiparticle phase space at high-energies*, *Comp. Phys. Comm.* **40** (1986) 359.

[13] A. van Hameren, R. Kleiss and P. D. Draggiotis, *SARGE: An algorithm for generating QCD-antennas*, *Phys. Lett. B* **483** (2000) 124–130.

[14] G. P. Lepage, "VEGAS: An adaptive multidimensional integration program." Cornell University, 1980. Publication CLNS-80/447.

[15] S. Donnachie, G. Dosch, P. Landshoff and O. Nachtmann, *Pomeron Physics and QCD*. No. 19 in Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology. Cambridge University Press, 2002.

[16] R. S. Thorne, *Parton distributions*, *Int. J. Mod. Phys.* **A19** (2004) 1074–1088 [hep-ph/0309343].

[17] L. N. Lipatov, *The parton model and perturbation theory*, *Sov. J. Nucl. Phys.* **20** (1975) 94–102.

[18] V. N. Gribov and L. N. Lipatov, *e^+e^- pair annihilation and deep inelastic e p scattering in perturbation theory*, *Yad. Fiz.* **15** (1972) 1218–1237.

[19] G. Altarelli and G. Parisi, *Asymptotic freedom in parton language*, *Nucl. Phys.* **B126** (1977) 298.

[20] Y. L. Dokshitzer, *Calculation of the structure functions for deep inelastic scattering and e^+e^- annihilation by perturbation theory in quantum chromodynamics. (in russian)*, *Sov. Phys. JETP* **46** (1977) 641–653.

[21] **UA5** Collaboration, G. J. Alner *et. al.*, *The UA5 high-energy anti-p p simulation program*, *Nucl. Phys.* **B291** (1987) 445.

[22] I. Borozan and M. Seymour, *An eikonal model for multiparticle production in hadron-hadron interactions*, *JHEP* **0209** (2002) 015.

[23] T. Sjöstrand and M. van Zijl, *A multiple-interaction model for the event structure in hadron collisions*, *Phys. Rev. D* **36** (1987) 2019–2041.

[24] G. Corcella, I. Knowles, G. Marchesini, S. Moretti, K. Odagiri, P. Richardson, M. Seymour and B. Webber, *HERWIG 6: An event generator for hadron emission reactions with interfering gluons (including supersymmetric processes)*, *JHEP* **0101** (2001) 010 [hep-ph/0011363].

[25] T. Sjöstrand, L. Lönnblad, S. Mrenna and P. Skands, *Pythia 6.3 physics and manual*, hep-ph/0308153.

[26] S. Frixione and B. Webber, *Matching NLO QCD computations and parton showers in heavy quark production*, *JHEP* **0206** (2002) 029 [hep-ph/0204244].

[27] S. Catani, F. Krauss, R. Kuhn and B. R. Webber, *QCD matrix elements + parton showers*, *JHEP* **0111** (2001) 063 [hep-ph/0109231].

[28] G. Corcella and M. Seymour, *Matrix element corrections to parton shower simulations of heavy quark decay*, *Phys. Lett. B* **442** (1998) 417–426 [hep-ph/9809451].

[29] R. Field and R. Feynman, *Quark elastic scattering as a source of high transverse momentum mesons*, *Phys. Rev.* **D15** (1977).

[30] G. Marchesini, L. Trentadue and G. Veneziano, *Space-time description of color screening via jet calculus techniques*, *Nucl. Phys.* **B181** (1980) 335.

[31] J.-C. Winter, F. Krauss and G. Soff, *A modified cluster-hadronization model*, hep-ph/0311085.

[32] T. Gleisberg, S. Hoeche, F. Krauss, A. Schaelicke, S. Schumann and J. Winter, "SHERPA 1.α, a proof-of-concept version." http://www.physik.tu-dresden.de/~krauss/hep/.

[33] **Particle Data Group** Collaboration, D. Groom *et. al.*, *Review of particle physics*, *Eur. Phys. J.* **C15** (2000) 1–878.

[34] P. Richardson, *Spin correlations in Monte Carlo simulations*, *JHEP* **0111** (2001) 029 [hep-ph/0110108].

[35] *CHEP Proceedings*, 1998. http://hep.ucsb.edu/people/lange/EvtGen/.

[36] S. Gieseke, P. Stephens and B. Webber, *New formalism for QCD parton showers*, *JHEP* **0312** (2003) 045 [hep-ph/0310083].

[37] S. Catani, B. R. Webber and G. Marchesini, *QCD coherent branching and semiinclusive processes at large x*, *Nucl. Phys. B* **349** (1991).

[38] S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock and B. R. Webber, *New clustering algorithm for multi-jet cross-sections in e^+e^- annihilation*, *Phys. Lett. B* **269** (1991).

[39] S. Catani, S. Dittmaier and Z. Trocsanyi, *One-loop singular behaviour of QCD and SUSY QCD amplitudes with massive partons*, *Phys. Lett. B* **500** (2001) [hep-ph/0011222].

[40] M. Cacciari and S. Catani, *Soft-gluon resummation for the fragmentation of light and heavy quarks at large x*, hep-ph/0107138.

[41] S. Gieseke, A. Ribon, M. Seymour, P. Stephens and B. Webber, *Herwig++ 1.0, physics and manual*, . preprint Cavendish-HEP-03/20, in preparation.

[42] S. Gieseke, A. Ribon, M. Seymour, P. Stephens and B. Webber, *Herwig++ 1.0, an event generator for e^+e^- annihilation*, *JHEP* **0402** (2004) 005 [hep-ph/0311208].

[43] S. Frixione and B. R. Webber, *Matching NLO QCD computations and parton shower simulations*, *JHEP* **0206** (2002) 029 [hep-ph/0204244].

[44] S. Frixione, P. Nason and B. R. Webber, *Matching NLO QCD and parton showers in heavy flavour production*, *JHEP* **0308** (2003) 007 [hep-ph/0305252].

[45] S. Frixione and B. R. Webber, *The MC@NLO 2.2 event generator*, hep-ph/0309186.

[46] Y. L. Dokshitzer, G. Marchesini and B. R. Webber, *Dispersive approach to power-behaved contributions in QCD hard processes*, *Nucl. Phys. B* **469** (1996) [hep-ph/9512336].

[47] G. Marchesini and B. R. Webber, *Simulation of QCD jets including soft gluon interference*, *Nucl. Phys. B* **238** (1984) 1.

[48] G. Marchesini and B. R. Webber, *Simulation of QCD coherence in heavy quark production and decay*, *Nucl. Phys. B* **330** (1990).

[49] Y. L. Dokshitzer, V. A. Khoze and S. I. Troian, *On specific QCD properties of heavy quark fragmentation ('dead cone')*, *J. Phys. G* **17** (1991).

[50] M. A. Aivazis, J. C. Collins, F. I. Olness and W. K. Tung, *Leptoproduction of heavy quarks. 2. a unified QCD formulation of charged and neutral current processes from fixed target to collider energies*, *Phys. Rev. D* **50** (1994) [hep-ph/9312319].

[51] R. S. Thorne and R. G. Roberts, *An ordered analysis of heavy flavour production in deep inelastic scattering*, *Phys. Rev. D* **57** (1998) [hep-ph/9709442].

[52] P. Nason and B. R. Webber, *Scaling violation in e^+e^- fragmentation functions: QCD evolution, hadronization and heavy quark mass effects*, *Nucl. Phys. B* **421** (1994). Erratum-ibid. B **480** (1994) 755.

[53] P. Nason and B. R. Webber, *Non-perturbative corrections to heavy quark fragmentation in e^+e^- annihilation*, *Phys. Lett. B* **395** (1997) [hep-ph/9612353].

[54] M. H. Seymour, *A simple prescription for first order corrections to quark scattering and annihilation processes*, *Nucl. Phys. B* **436** (1995) [hep-ph/9410244].

[55] G. Corcella and M. H. Seymour, *Initial state radiation in simulations of vector boson production at hadron colliders*, *Nucl. Phys. B* **565** (2000) [hep-ph/9908388].

[56] G. Corcella and M. H. Seymour, *Matrix element corrections to parton shower simulations of heavy quark decay*, *Phys. Lett. B* **442** (1998) [hep-ph/9809451].

[57] A. Kupco, *Cluster hadronization in herwig 5.9*, hep-ph/9906412.

[58] I. G. Knowles and G. D. Lafferty, *Hadronization in Z^0 decay*, *J. Phys. G* **23** (1997) [hep-ph/9705217].

[59] B. R. Webber, *Fragmentation and hadronization*, *Int. J. Mod. Phys.* **A15S1** (2000) 577–606 [hep-ph/9912292].

[60] **OPAL** Collaboration, K. Ackerstaff *et. al.*, *Measurements of flavour dependent fragmentation functions in $Z^0 \to q\bar{q}$ events*, *Eur. Phys. J.* **C7** (1999) 369–381 [hep-ex/9807004].

[61] **OPAL** Collaboration, R. Akers *et. al.*, *Measurement of the production rates of charged hadrons in e^+e^- annihilation at the Z^0*, *Z. Phys.* **C63** (1994) 181–196.

[62] G. Corcella and *et al.*, *HERWIG 6.5 release note*, hep-ph/0210213.

[63] M. Bertini, L. Lönnblad and T. Sjöstrand, *Pythia version 7-0.0: A proof-of-concept version*, *Comput. Phys. Commun.* **134** (2001) [hep-ph/0006152].

[64] http://http://www.thep.lu.se/ThePEG/.

[65] L. Lönnblad, *CLHEP: A project for designing a C++ class library for high-energy physics*, *Comput. Phys. Commun.* **84** (1994). http://wwwasd.web.cern.ch/wwwasd/lhc++/clhep/.

[66] B. Stroustrup, *The C++ Programming Language*. Addison-Wesley, 3rd ed., 2000.

[67] L. Lönnblad. Private communication.

[68] M. Glück, E. Reya and A. Vogt, *Dynamical parton distributions of the proton and small x physics*, *Z. Phys.* **C67** (1995) 433–448.

[69] A. Martin, R. Roberts, W. Stirling and R. Thorne, *MRST2001: partons and α_s from precise deep inelastic scattering and Tevatron jet data*, *Eur. Phys. J.* **C23** (2002) 73–87 [hep-ph/0110215].

[70] J. Anderson. Private communication.

[71] R. Kleiss, *From two to three jets in heavy boson decays: an algorithmic approach*, *Phys. Lett. B* **180** (1986).

[72] S. Catani, S. Dittmaier, M. H. Seymour and Z. Trocsanyi, *The dipole formalism for next-to-leading order QCD calculations with massive partons*, *Nucl. Phys. B* **627** (2002) [hep-ph/0201036].

[73] M. H. Seymour, *Matrix element corrections to parton shower algorithms*, *Comput. Phys. Commun.* **90** (1995) [hep-ph/9410414].

[74] B. R. Webber, *A QCD model for jet fragmentation including soft gluon interference*, *Nucl. Phys. B* **238** (1984).

[75] P. D. Acton *et. al.*, *A study of charged particle multiplicities in hadronic decays of the Z^0*, *Z. Phys.* **C53** (1992).

[76] Y. L. Dokshitzer, *Hard QCD working group: Theory summary*, *J. Phys. G* **17** (1991). and contribution at *Workshop on Jet Studies at LEP and HERA*, Durham, December 1990, cited in W. J. Stirling.

[77] J. M. Butterworth, J. P. Couchman, B. E. Cox and B. M. Waugh, *KtJet: A C++ implementation of the $K_\perp$ clustering algorithm*, *Comput. Phys. Commun.* **153** (2003) [hep-ph/0210022].

[78] **JADE** Collaboration, P. Pfeifenschneider *et. al.*, *QCD analyses and determinations of α_s in e^+e^- annihilation at energies between 35 GeV and 189 GeV*, *Eur. Phys. J.* **C17** (2000) [hep-ex/0001055].

[79] **JADE** Collaboration, P. A. M. Fernandez, O. Biebel, S. Bethke, S. Kluth and P. Pfeifenschneider, *A study of event shapes and determinations of α_s using data of e^+e^- annihilations at $\sqrt{s} = 22$ GeV to 44 GeV*, *Eur. Phys. J.* **C1** (1998) [hep-ex/9708034].

[80] S. Catani, Y. L. Dokshitzer, F. Fiorani and B. R. Webber, *Average multiplicities in two and three jet e^+e^- annihilation events*, *Nucl. Phys. B* **383** (1992).

[81] **DELPHI** Collaboration, P. Abreu *et. al.*, *Tuning and test of fragmentation models based on identified particles and precision event shape data*, *Z. Phys.* **C73** (1996) 11–60.

[82] S. Catani, G. Turnock and B. R. Webber, *Jet broadening measures in e^+e^- annihilation*, *Phys. Lett.* **B295** (1992) 269–276.

[83] S. J. Burby and E. W. N. Glover, *Resumming the light hemisphere mass and narrow jet broadening distributions in e^+e^- annihilation*, *JHEP* **04** (2001) 029 [hep-ph/0101226].

[84] R. Kuhn, F. Krauss, B. Ivanyi and G. Soff, *APACIC++ 1.0: A parton cascade in C++*, *Comput. Phys. Commun.* **134** (2001) [hep-ph/0004270].

[85] H. Hoeth, *Messung der Vierjet–Winkelverteilungen und Bestimmung der QCD–Farbfaktoren mit Hilfe des APACIC++–Generators*. PhD thesis, Fachbereich Physik, Bergische Universität Wuppertal, 2003. WUD 03-11.

[86] **SLD** Collaboration, K. Abe *et. al.*, *Production of π^+, π^-, K^+, K^-, p and $\bar{p}$ in light (uds), c and b jets from Z^0 decays*, *Phys. Rev.* **D69** (2004) 072003 [hep-ex/0310017].

[87] **ALEPH** Collaboration, R. Barate *et. al.*, *Studies of quantum chromodynamics with the ALEPH detector*, *Phys. Rept.* **294** (1998) 1–165.

[88] **SLD** Collaboration, K. Abe *et. al.*, *Measurement of the b-quark fragmentation function in Z^0 decays*, *Phys. Rev.* **D65** (2002), no. 092006 [hep-ex/0202031]. [Erratum-ibid. D **66** (2002) 079905].

[89] **ALEPH** Collaboration, A. Heister *et. al.*, *Study of the fragmentation of b quarks into B mesons at the Z^0 peak*, *Phys. Lett.* **B512** (2001) [hep-ex/0106051].

[90] S. Martin, *A supersymmetry primer*, hep-ph/9709356.

[91] S. Weinberg, *The Quantum Theory of Fields*, vol. III. Cambridge University Press, Cambridge, UK, 2000.

[92] H. E. Haber, *Introductory low-energy supersymmetry*, hep-ph/9306207.

[93] ATLAS Collaboration, *Detector and Physics Performance TDR*, 1999. CERN/LHCC.

[94] CMS Collaboration, *Technical Proposal*. http://cmsinfo.cern.ch/TP/TP.html.

[95] I. Hinchliffe and P. Richardson, *Supersymmetric models and collider signatures*, *J. Phys. G* **27** (2001) 2485–2496.

[96] A. Barr, C. Lester and P. Stephens, *m_T^2: The truth behind the glamour*, *J. Phys.* **G29** (2003) 2343–2363 [hep-ph/0304226].

[97] H. Euler and W. Heisenberg, *Consequences of Dirac's theory of positrons*, *Z. Phys.* **98** (1936) 714.

[98] J. Schwinger, *On gauge invariance and vacuum polarization*, *Phys. Rev.* **82** (1951) 664–679.

[99] J. Goldstone, A. Salam and S. Weinberg, *Broken symmetries*, *Phys. Rev.* **127** (1962) 965–970.

[100] M. Quirós, *Finite temperature field theory and phase transitions*, hep-ph/9901312.

[101] W. Siegel, *Supersymmetric dimensional regularization via dimensional reduction*, *Phys. Lett.* **B84** (1979) 193.

[102] S. Weinberg, *Perturbative calculations of symmetry breaking*, *Phys. Rev. D* **7** (1973) 2887–2910.

[103] G. Passarino and M. J. G. Veltman, *One loop corrections for e^+e^- annihilation into $\mu^+\mu^-$ in the Weinberg model*, *Nucl. Phys.* **B160** (1979) 151.

[104] A. Denner, *Techniques for calculation of electroweak radiative corrections at the one loop level and results for W physics at LEP-200*, *Fortschr. Phys.* **41** (1993) 307–420.

[105] D. M. Pierce, J. A. Bagger, K. T. Matchev and R.-j. Zhang, *Precision corrections in the minimal supersymmetric standard model*, *Nucl. Phys.* **B491** (1997) 3–67 [hep-ph/9606211].

[106] R. Mertig, M. Bohm and A. Denner, *FeynCalc: Computer algebraic calculation of Feynman amplitudes*, *Comput. Phys. Commun.* **64** (1991) 345–359.

[107] T. Hahn and M. Perez-Victoria, *Automatized one-loop calculations in four and d dimensions*, *Comput. Phys. Commun.* **118** (1999) 153–165 [hep-ph/9807565].

[108] J. Callan, Curtis G., *Broken scale invariance in scalar field theory*, *Phys. Rev.* **D2** (1970) 1541–1547.

[109] K. Symanzik, *Small distance behavior in field theory and power counting*, *Commun. Math. Phys.* **18** (1970) 227–246.

[110] N. Cabibbo, L. Maiani, G. Parisi and R. Petronzio, *Bounds on the fermions and Higgs boson masses in grand unified theories*, *Nucl. Phys.* **B158** (1979) 295.

[111] J. R. Ellis, S. Kelley and D. V. Nanopoulos, *Probing the desert using gauge coupling unification*, *Phys. Lett.* **B260** (1991) 131–137.

[112] C. Ford, D. R. T. Jones, P. W. Stephenson and M. B. Einhorn, *The effective potential and the renormalization group*, *Nucl. Phys.* **B395** (1993) 17–34 [hep-lat/9210033].

[113] J. R. Forshaw, A. Sabio Vera and B. E. White, *Mass bounds in a model with a triplet Higgs*, *JHEP* **06** (2003) 059 [hep-ph/0302256].

[114] J. Hetherington, *Phenomenology of Supersymmetric Models*. PhD thesis, University of Cambridge, November, 2002.

[115] C. Bauer, A. Frink and R. Kreckel, *Introduction to the GiNaC framework for symbolic computation within the C++ programming language*, *J. Symbolic Computation* **33** (2002) 1–12.

[116] B. C. Allanach, *SOFTSUSY: A C++ program for calculating supersymmetric spectra*, *Comput. Phys. Commun.* **143** (2002) 305–331 [hep-ph/0104145].